

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Rozpoznávání částí lidského těla v hloubkových mapách

Recognising the Parts of Human Bodies in the Depth Maps

Zadání diplomové práce

Student:

Bc. Patrik Klement

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Rozpoznávání částí lidského těla v hloubkových mapách
Recognising the Parts of Human Bodies in the Depth Maps

Jazyk vypracování:

čeština

Zásady pro vypracování:

Detekce postav (a jejich částí) v obraze je významnou úlohou, kterou lze uplatnit v různých aplikacích (čítání osob, vyhodnocování aktivit atd). Detekce z hloubkových map může být výhodnější než detekce z RGB obrazů, protože barva (a tím i spolehlivost rozpoznání) může být velmi podstatně ovlivněna texturou oděvu. V současné době jsou na trhu hloubkové senzory poskytující hloubkové mapy v dostatečné kvalitě. V diplomové práci proveďte následující:

1. V rešeršní části práce stručně charakterizujte několik již známých postupů. Můžete akcentovat přístupy, jejichž výsledkem je stanovení kostry.
2. Navrhněte vlastní metodu detekce kostry nebo implementujte některou z metod již dříve popsanych.
3. Metodu důkladně experimentálně ověřte. Popište její problémy a přednosti.
4. Dosazené výsledky řádně zdokumentujte v textové části práce. Implementaci proveďte v C/C++.

Seznam doporučené odborné literatury:

Julio Cezar Silveira Jacques et al.: Skeleton-based human segmentation in still images, ICIP 2012

Souto, H. and Musse, S.: Automatic Detection of 2D Human Postures Based on Single Images, In Proc. 24th SIBGRAP, 2011, pp 48-55

Wang, J. et al.: Learning Actionlet Ensemble for 3D Human Action Recognition. IEEE TPAMI. 2014 May; 36(5), pp 914-27.

a další

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Dr. Ing. Eduard Sojka**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017




doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 13. dubna 2017


.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 13. dubna 2017



.....

Rád bych na tomto místě poděkoval vedoucímu doc. Dr. Ing. Eduardu Sojkovi za jeho cenné rady a zkušenosti při realizaci této práce. Dále bych chtěl poděkovat rodině, přítelkyni a přátelům za jejich podporu v nejtěžších chvílích.

Abstrakt

Cílem této práce je rozpoznávání lidských částí těla v hloubkové mapě, která je získána pomocí zařízení zvané Kinect v1 od firmy Microsoft. Práce je realizována v jazyce C++ s využitím knihovny OpenCV, která je určená pro práci s obrazem. Při realizaci této práce je zapotřebí získat obraz z hloubkové mapy. Ten je získán za pomoci několika různých metod, které budou důkladně popsány v tomto dokumentu. Následně v hloubkové mapě budou nalezeny části lidského těla jako je hlava, ruce a nohy. V první části si probereme základní informace k tématu a technologie využitě v této práci. Dále se také podíváme na již dosažené výsledky a způsoby při realizaci tohoto problému od jiných autorů. Ve druhé části se podíváme na vlastní průběh práce, její implementaci a další informace spojené s tvorbou diplomové práce.

Klíčová slova: Kinect, hloubková mapa, části těla, OpenCV, detekce

Abstract

The aim of this thesis is to recognize parts of human body in depth map which is obtained by a device called Kinect v1 from Microsoft company. The thesis is implemented in C++ programming language with the use of OpenCV library which is intended for the work with image. In the implementation of this thesis is necessary to get a picture from depth map. This is obtained by using several different methods that are thoroughly described in this document. Subsequently, in the depth map will be found human body parts such as the head, hands and feet. In the first part, we will describe the basic information associated with this topic and technologies which are used in this work. Additionally, we will also describe existing solutions and ways to realize this topic from another authors. In the second part, we will look at the own workflow, its implementation and other information related to the diploma thesis.

Key Words: Kinect, depth map, body parts, OpenCV, detection

Obsah

Seznam použitých zkratk a symbolů	10
Seznam obrázků	11
Seznam tabulek	13
1 Úvod	14
2 Základní informace	16
2.1 Jazyk C++	16
2.2 Kinect v1 a Kinect SDK	16
2.3 OpenCV	17
2.4 Hlubková mapa	17
2.5 Prahování	18
2.6 Indexování obrazu	18
3 Existující řešení	20
3.1 Detekce osob a sledování hlavy	20
3.2 Real-time rozpoznání lidských póz v hloubkových obrazech	21
3.3 Přesné odhady 3D póz z hloubkové mapy	22
3.4 Tvorba 3D modelu řidiče na základě hloubkových map	23
4 Vlastní postupy a využití metody	24
4.1 Inicializace a získání dat z Kinectu	25
4.2 Odstranění pozadí a získání obrysu člověka	25
4.3 Odstranění nežádoucích objektů	28
4.4 Nalezení hlavy	30
4.5 Nalezení ramen	34
4.6 Nalezení kyčlí	35
4.7 Nalezení loktů a rukou	37
4.8 Nalezení kolen a nohou	40
5 Experimentální část	43
5.1 Datová sada	43
5.2 Výsledky detekce	45
5.3 Špatné natočení hlavy	48
5.4 Vyhodnocení úspěšnosti	49
5.5 Možná vylepšení	51

6 Závěr	53
Literatura	54
Přílohy	55
A Obsah přiloženého CD	56

Seznam použitých zkratk a symbolů

OpenCV	– Multiplatformní knihovna pro manipulaci s obrazem
RGB	– Barevný model červená-zelená-modrá (red-green-blue)
IR	– Infračervený obraz (Infrared)
SDK	– Software Development Kit
v1, v2	– Verze 1, 2
3D	– Trojrozměrný
OS X	– Operační systém pro počítače Macintosh
API	– Application Programming Interface - rozhraní pro programování aplikací

Seznam obrázků

1	Jednotlivé typy obrazu, které lze získat z kamery.	14
2	Nalezení lidské kostry v hloubkové mapě pomocí navrženého algoritmu.	15
3	Nejpoužívanější programovací jazyky světa.	16
4	Pohybový senzor Kinect (vlevo) a popis jeho částí (vpravo).	17
5	Bimodální histogram jasu.	18
6	Princip indexování objektů.	19
7	Vymezení osoby používáním referenční hloubky obrazu. Nahoře se nachází RGB snímky, uprostřed snímky hloubkové mapy a dole obrázky po získání obrysu člověka.	20
8	Detekce hlavy v hloubkové mapě za pomoci filtrování částic.	21
9	Znázornění jednotlivých částí lidského těla po přiřazení pixelů do dané oblasti.	22
10	Znázornění snímků lidské kostry, které jsou uloženy v databázi pro následné porovnávání, včetně 19 spojení označených pomocí červených koleček	22
11	Ukázka výsledků detekce lidské kostry na základě hloubkových map za pomoci navrženého algoritmu autorem diplomové práce.	23
12	Senzor Intel RealSense R200.	24
13	Rozdíl získaného obrazu mezi Kinect v1 a Kinect v2.	24
14	Získání jednoho snímku z Kinectu.	25
15	Porovnání vstupu a výsledku po prahování.	26
16	Rozdíl jednotlivých algoritmů pro odstranění pozadí.	27
17	Princip metody pro odstranění pozadí, která je použita v programu.	28
18	Vstupní obraz pro zaindexování (vlevo) a výsledek po provedení (vpravo).	29
19	Výsledek po odstranění nežádoucích objektů pomocí indexování a nalezení objektu s největším obvodem.	30
20	Metoda č.1 pro nalezení hlavy, která je založena na vyhledání dvou bodů nahoře a dvou bodů dole.	31
21	Metoda č.2 pro nalezení hlavy, která funguje na principu nalezení krajních bodů na všech stranách hlavy.	32
22	Šablona pro porovnání při detekci hlavy v obraze.	33
23	Výsledek metody pro nalezení hlavy pomocí šablony (vlevo) a zobrazení nalezené hlavy pomocí červeného čtverce (vpravo).	34
24	Nalezení ramen lidského těla v hloubkové mapě za pomoci krajních bodů hlavy, posunutí do stran od těchto bodů a následného posunutí směrem dolů, až narazíme na ramena.	35
25	Znázornění vypočtených normál ze středu ramen.	36
26	Výsledné nalezení kyčlí ve spodní části trupu za pomoci výpočtů normál ze středu ramen směrem dolů.	37

27	Princip nalezení loktů za pomoci bodů na kružnici (vlevo) a zobrazení výsledné detekce loktů (vpravo).	38
28	Znázornění výpočtu vzdálenosti mezi ramenem a loktem pomocí Pythagorovy věty (vlevo) a princip nalezení rukou pomocí bodů na kružnici (vpravo).	39
29	Výsledná detekce rukou za pomoci procházení bodů na kružnici.	40
30	Princip nalezení kolen s využitím bodů na kružnici (vlevo) a výsledná detekce kolen za pomoci této metody (vpravo).	41
31	Znázornění výpočtu vzdálenosti mezi kyčlí a kolenem za pomoci Pythagorovy věty (vlevo) a princip nalezení chodidel při procházení bodů (vpravo).	42
32	Výsledek nalezení nohou za použití algoritmu detekce bodů na kružnici.	42
33	Ukázka jednotlivých postojů, které jsou použity pro detekci lidské kostry v hloubkové mapě.	44
34	Přehled pozadí, které se vyskytovaly před senzorem během snímání a natáčení videí.	45
35	Výsledek detekce zvolených postojů po použití navrženého postupu.	46
36	Problémy při špatném natočení těla k senzoru.	47
37	Problémy při doteku ruky a nohy, které nastávají při detekci.	48
38	Chybné detekce hlavy vlivem nedodržení podmínek pro úspěšnou detekci pomocí šablony.	49
39	Příklad špatné a správné detekce hlavy ve Videu 1.	50
40	Příklad správné a špatné detekce celé lidské kostry ve Videu 3.	51
41	Znázornění úspěšných detekcí lidské kostry pořízené ve Videu 2.	52

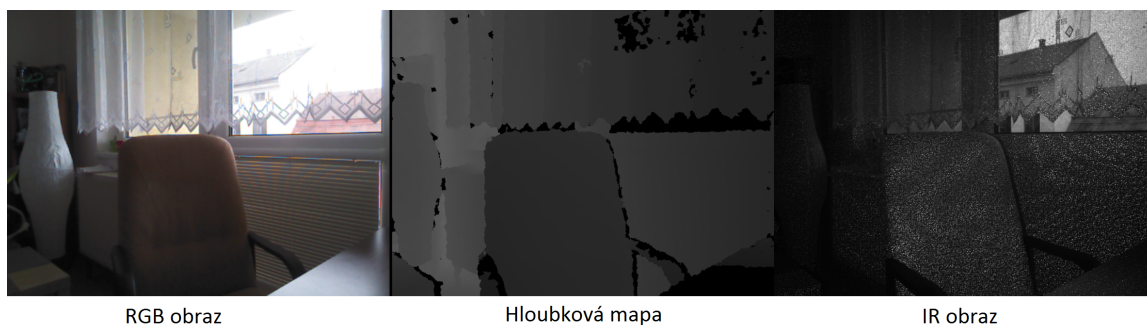
Seznam tabulek

1	Úspěšnost algoritmu při detekci hlavy.	50
2	Úspěšnost detekce lidského těla při použití algoritmu.	51

1 Úvod

Počítačová grafika a strojové vidění je v dnešní době velmi atraktivní téma. Může usnadnit spoustu věcí a naskytuje se možnost automatizace. Zpracování obrazu může být využito v širokém odvětví jako je například bezpečnost, továrny a jiné.

V dnešní době se všude kolem nás vyskytuje velké množství kamer, které slouží například pro kontrolu bezpečnosti ve světě. Tyto kamery mohou být vybaveny více možnostmi, jak snímat okolní svět. První z možností je především velmi dobře známý barevný obraz, který se nazývá RGB. Vyskytuje se všude kolem nás, například v televizi, mobilech a dalších zařízeních. Méně známý je však infračervený, který se využívá především ve tmavých místnostech bez možnosti přisvícení nebo venku ve tmě jako hlídání objektu. Jako poslední a pro nás zajímavá část pro tuto práci je hloubková mapa. Tato možnost obrazu nemusí být všem příliš známá, a proto si ji popíšeme v následující kapitole, abychom mohli plně vstoupit do hloubky tohoto tématu a porozumět mu.



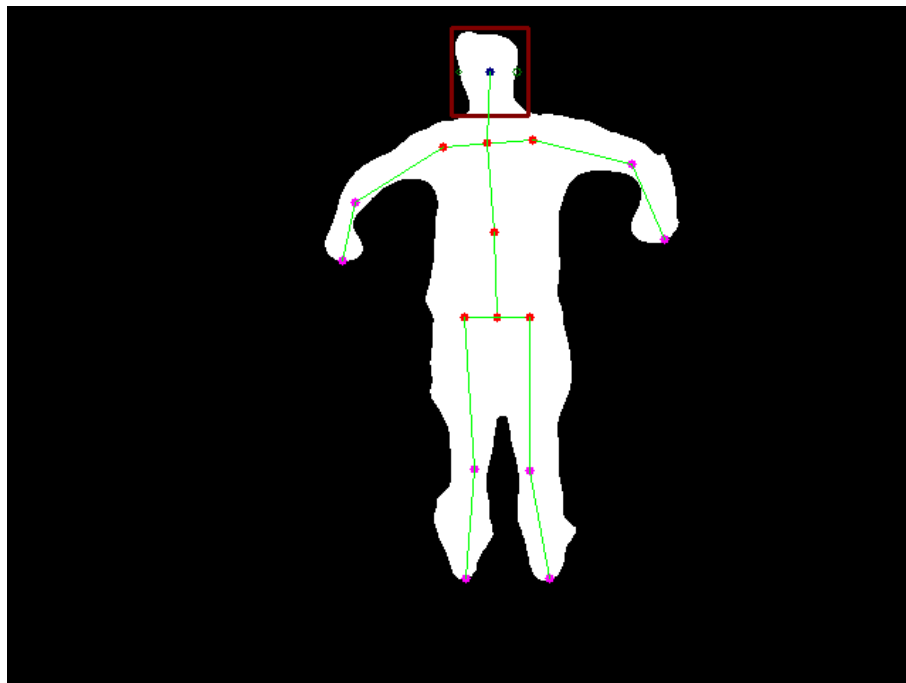
Obrázek 1: Jednotlivé typy obrazu, které lze získat z kamery.

Hloubková mapa může být také využita například ve vozidlech pro snímání řidiče. Určitě si řekneme, že v autě mohou být různé světelné podmínky, řidič může být stále osvětlován z protijedoucích aut, a tudíž by v tomto případě RGB kamera neměla příliš význam, protože by mohla zkreslovat konečné výsledky programu a mohlo by dojít ke špatnému vyhodnocování. Z tohoto důvodu je dobré využití hloubkové mapy, která těmito faktory není ovlivněna.

V této práci se tedy podíváme na detekci lidské kostry v hloubkové mapě, která je na Obrázku 2. Jedná se o nalezení hlavy, částí ramen, ruce a nohy. Toto téma může být využito v širokém odvětví, jako je například detekce akcí člověka, jako je chůze, běh, sezení, dřepění nebo ve výše zmíněném vozidle jako kontrola řidiče, zdali má ruce na volantě nebo mu upadá hlava vlivem mikrosnánky a jiné. Využití hloubkové mapy může být také jako detekce člověka v prostoru a jeho pohyb. Práce je realizována pomocí jazyka C++ s využitím knihovny OpenCV a také pomocí zařízení zvané Kinect v1 a softwaru Kinect SDK. Všechny tyto technologie a zařízení budou později podrobně popsány.

V první části práce si popíšeme základní informace o nástrojích a technice, kterou jsme při této práci využili a podíváme se na již dosažené výsledky jiných autorů, jejich postupy a nápady,

které využili při realizaci tohoto tématu. Ve druhé části se podíváme na vlastní implementaci a také různé problémy při vytváření, postupy, nápady a různé metody, které vedly k úspěšnému vytvoření této práce.



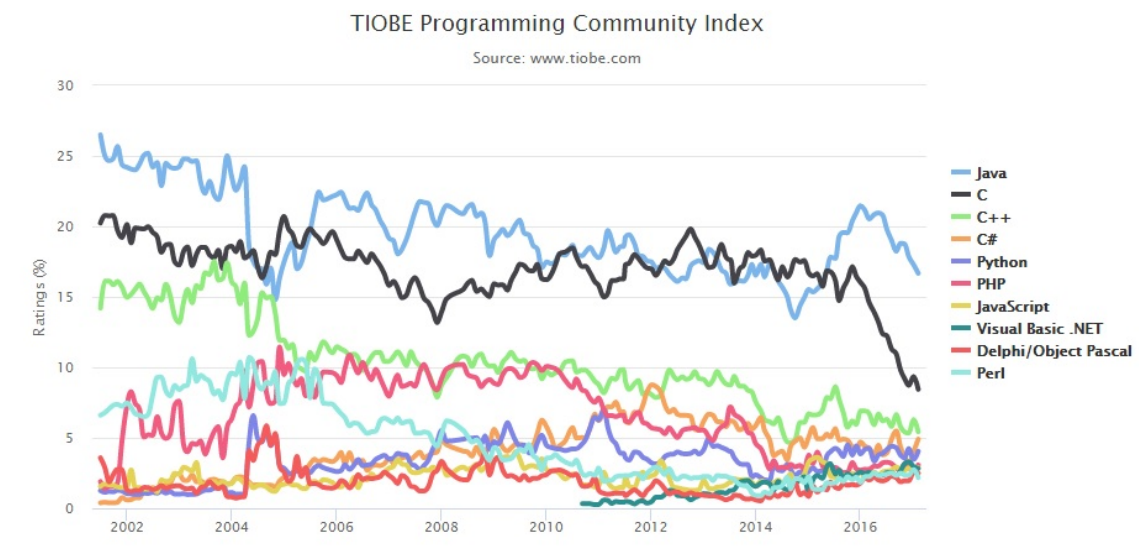
Obrázek 2: Nalezení lidské kostry v hloubkové mapě pomocí navrženého algoritmu.

2 Základní informace

V této části si popíšeme základní informace, bez kterých bychom se v této práci neobešli. Jedná se o programovací jazyk, využití knihovny a také nástroje, které byly použity. Dále si řekneme, co to vlastně hloubková mapa je, abychom lépe porozuměli tématu a popíšeme si princip metod, které byly v této práci využity.

2.1 Jazyk C++

C++ je multiparadigmatický programovací jazyk, který vzniknul na základě rozšíření jazyka C. Vzniknul v roce 1983 a jeho autorem je Bjarne Stroustrup. Podporuje různé styly programování jako je procedurální programování, objektově orientované programování a generické programování [1]. Jazyk C++ je podle **Tiobe indexu** třetím nejrozšířenějším programovacím jazykem na světě [2].



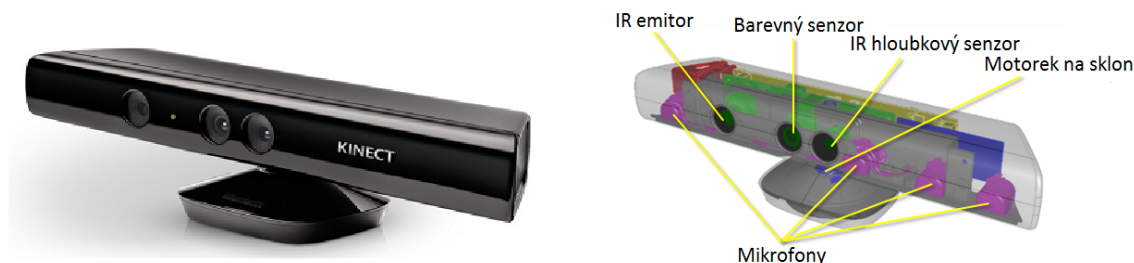
Obrázek 3: Nejpoužívanější programovací jazyky světa.

2.2 Kinect v1 a Kinect SDK

Microsoft Kinect je 3D senzor, který byl původně vytvořen pro zařízení Xbox. V herním průmyslu si toto zařízení našlo velmi dobré využití z důvodu interakce s uživatelem, kdy hráč je schopen ovládat hru svými pohyby nebo hlasovými příkazy. V naší práci však o hry nejde. První generace Kinectu byla představena v listopadu 2010 pro zařízení Xbox. Verze pro Windows byla představena v únoru roku 2012 [3]. Díky tomu, že byla možnost Kinect připojit i k počítači se otevřela velká škála možností, jak tento přístroj využít, například i ve zpracování obrazu.

Je to tedy zařízení, které disponuje třemi předními kamerami, které lze vidět na Obrázku 4. První kamera zleva je IR emitor. Ten se stará o vyzařování infračervených paprsků do prostoru,

které následně kamera zcela vpravo čte. Ta se nazývá IR hloubkový senzor a odražené paprsky převádí na hloubkovou informaci, tedy vzdálenost mezi objektem a senzorem [4]. Kamera uprostřed se stará o zachycení barevného obrazu. Dále se ještě na senzoru vyskytují mikrofony, které mohou sloužit k hlasovému ovládání.



Zdroj: <https://msdn.microsoft.com/en-us/library/hh855355.aspx> Zdroj: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>

Obrázek 4: Pohybový senzor Kinect (vlevo) a popis jeho částí (vpravo).

Dále jsem se pro práci s Kinectem rozhodl využít **Microsoft Kinect SDK** z důvodu jednoduché instalace a spolupráce s vývojovým prostředím Visual Studio. Jedinou nevýhodou je, že je pouze pro systémy Windows a lze jej použít s programovacími jazyky C, C++, C# a Visual Basic.

2.3 OpenCV

OpenCV (Open Source Computer Vision) je knihovna programovacích funkcí, které jsou zaměřeny především na zpracování obrazu. Původně bylo OpenCV vyvíjeno firmou **Intel**. OpenCV bylo vyvinuto za účelem zajistit společnou infrastrukturu pro aplikace počítačového vidění a urychlit zpracování obrazu. Je napsán primárně v jazyce C++ a běží na různých platformách, jako jsou například Windows, Linux, OS X a další [5].

Knihovna má více než 2500 algoritmů. Algoritmy mohou být využity například pro detekci a rozpoznávání obličejů, různých objektů, rozpoznávání lidských pohybů a akcí z videa a mnoho dalších.

2.4 Hloubková mapa

Abychom se mohli hlouběji podívat do tématu, musíme si nejprve říct, co to vlastně ta hloubková mapa je. Hloubková mapa je tedy snímek, ve kterém každý bod představuje vzdálenost od senzoru, která je udávána v milimetrech. Námi používaný Kinect podporuje dva režimy snímání [6]:

- Výchozí - rozsah snímané vzdálenosti je 80 až 400cm,
- Near mode - rozsah 50 až 300cm.

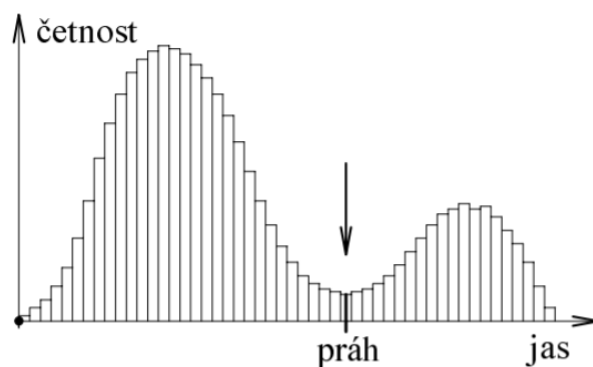
Na Obrázku 4 lze vidět tři senzory, z nichž pro hloubkovou mapu využíváme první zleva a první vpravo. Jedná se o IR emitor a IR hloubkový senzor. Jak jsem již zmínil, výhodou infračerveného světla je, že nejde ve tmě vidět, avšak může být silně rušeno přímým sluncem.

2.5 Prahování

Prahování je funkce, která upravuje hodnoty na základě Předpisu 1 [14]. Je to tedy oddělení částí obrazu od sebe, kdy si stanovíme takzvaný práh. V nekomplikovaných případech je tato metoda považována za velmi rychlou a spolehlivou. Správná hodnota pro zvolení práhu lze určit pomocí histogramu, na kterém závisí úspěšnost prahování. Například pro obrazy s bimodálním histogramem (Obrázek 5) je dobré zvolit hodnotu, která se rovná minimu mezi oběma vrcholy [15]. Pro obraz, který je však více kontrastní lze využít prahování s proměnnou hodnotou práhu, abychom mohli využít prahování na různých místech obrazu úspěšně.

$$dst(x, y) = \begin{cases} maxVal & \text{if } src(x, y) > thresh \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Nyní pro každý pixel porovnáváme jeho hodnotu se zvoleným práhem. V případě, že je hodnota větší než práh bude nově nastavena nejvyšší hodnota. V našem případě je to hodnota 255 a v opačném případě bude nastavena hodnota 0. Při použití této funkce vycházíme z předpokladu, že jednotlivé hledané pixely mají stejný nebo dost podobný jas, aby bylo snadné je nalézt.



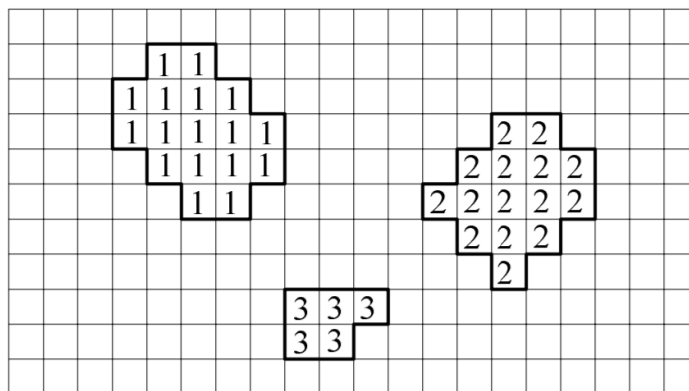
Zdroj: http://mrl.cs.vsb.cz/people/sojka/dzo/digitalni_zpracovani_obrazu.pdf

Obrázek 5: Bimodální histogram jasu.

2.6 Indexování obrazu

Indexování se využívá pro další zpracování binárního obrazu (v našem případě to jsou hodnoty 0 nebo 255) [15]. Indexování je tedy přiřazení přirozeného čísla do každého pixelu pro danou nalezenou oblast viz Obrázek 6. Za jednu oblast je považována množina sousedních pixelů, ve kterých je zapsána stejná hodnota. Při přiřazování hodnoty rozlišujeme dva typy:

1. **Čtyřsousednost** - tento typ znamená, že dva pixely jsou sousedící právě tehdy, mají-li společnou stranu. Z toho vyplývá, že každý pixel může mít až čtyři sousedy.
2. **Osmisousednost** - při tomto typu jsou sousedními pixely i ty, které sousedí hranou nebo pouze rohem. Z toho tedy vyplývá, že sousedních pixelů může být osm.



Zdroj: http://mrl.cs.vsb.cz/people/sojka/dzo/digitalni_zpracovani_obrazu.pdf

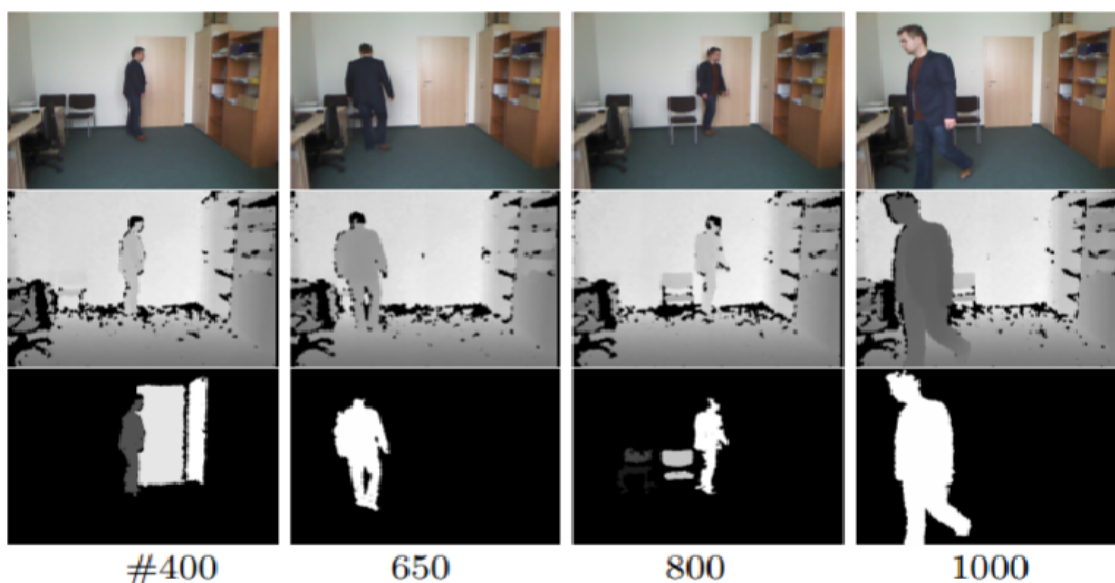
Obrázek 6: Princip indexování objektů.

3 Existující řešení

V této kapitole se podíváme na již existující řešení, která vedla k úspěšnému nalezení hlavy nebo celé lidské kostry jako celek. Toto téma je ve světě poněkud novější a nelze počítat s velkým množstvím prací, které vedou k téměř stoprocentní úspěšnosti.

3.1 Detekce osob a sledování hlavy

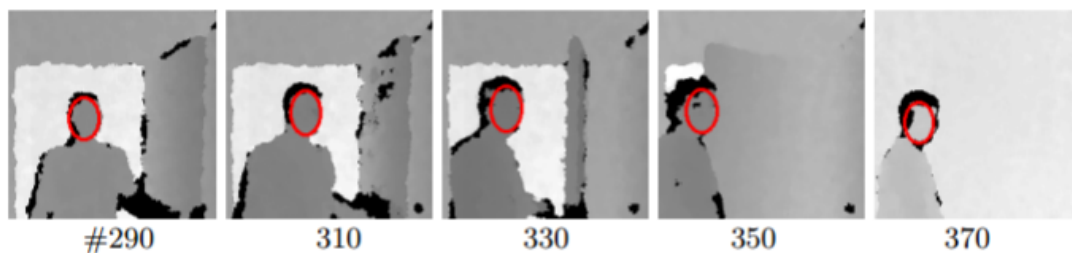
Jako první si předvedeme práci, na které se podíleli autoři Michal Kepski a Bogdan Kwalek [7]. Ti ve své práci nejdříve popisují detekci člověka v hloubkové mapě. Princip spočívá v tom, že se pro referenční obraz vypočítá pro každý pixel střední hodnota z několika předchozích snímků. Následně si výpočet hodnot uloží do seřazeného listu. Za pomoci odstranění nejstarších pixelů v listu a nahrazením seřazených hodnot z aktuální získané hloubkové mapy lze získat obrys lidského těla v hloubkové mapě. Pro úspěšnou detekci se po různých experimentech zdá jako dostatečný počet třináct po sobě jdoucích snímků pro výpočet střední hodnoty pro každý pixel. V případě, že se člověk nehýbe, nedojde k aktualizaci referenčního obrazu. Pro odstranění nežádoucích objektů je tedy využito porovnání aktuálního obrazu s referenčním. Výsledek tohoto postupu lze shlédnout na Obrázku 7.



Zdroj: http://galaxy.uci.agh.edu.pl/~bkw/research/pdf/2014/KepskiKwalek_ICCVG2014.pdf

Obrázek 7: Vymezení osoby používáním referenční hloubky obrazu. Nahoře se nachází RGB snímky, uprostřed snímky hloubkové mapy a dole obrázky po získání obrysu člověka.

Jako další je v práci výše uvedených autorů detekce hlavy. Detekce hlavy je za pomoci využití filtrování částic. Hlava je zobrazena za pomoci elipsoidu. Ta je zobrazena na základě výpočtů hodnot mezi elipsoidem a mračnem bodů, a dále mezi promítaným elipsoidem a hranami v hloubkové mapě. Přesný postup pro detekci hlavy a důležité výpočty pro detekci lze shlédnout v dokumentu autorů [7]. Výsledek detekce hlavy je zobrazen na Obrázku 8.



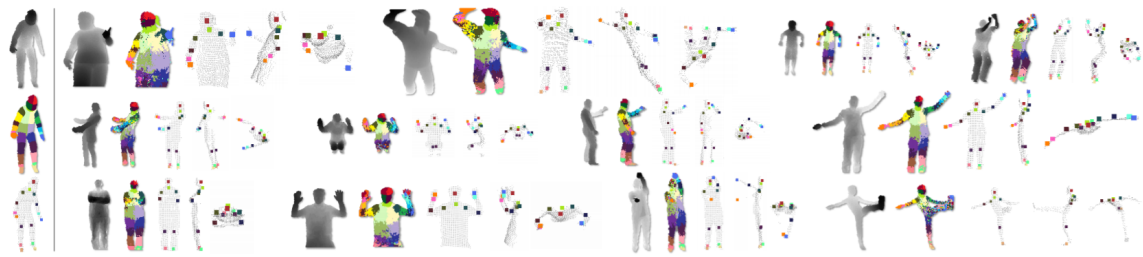
Zdroj: http://galaxy.uci.agh.edu.pl/~bkw/research/pdf/2014/KepskiKwolek_ICCVG2014.pdf

Obrázek 8: Detekce hlavy v hloubkové mapě za pomoci filtrování částic.

3.2 Real-time rozpoznání lidských póz v hloubkových obrazech

Další existující řešení je detekování 3D polohy těla a jednotlivých částí z hloubkové mapy [8]. Princip spočívá v tom, že se lidská postava rozdělí na několik částí, kdy se pro každý pixel v hloubkové mapě určí, do které části lidského těla patří. Dále máme předem připravené body, které jsou přiřazeny jednotlivým částem lidského těla a pro každý pixel z hloubkové mapy se určí, kde s největší pravděpodobností daný pixel náleží.

Abychom věděli, kam daný pixel zařadit, je zde využitý randomizovaný rozhodovací les, který se řídí podle hloubky v okolních místech. Jako trénovací data byly použity různé pozice lidského těla, které byly uloženy v databázi. Randomizovaný rozhodovací les byl trénován právě na těchto snímcích o velikosti až tisíců různých pozic. Tento algoritmus je schopen zpracovat až 200 snímků za vteřinu. Hlavní myšlenkou bylo rozpoznávat pózy lidského těla, například pro hraní her nebo ovládání různých zařízení. Z tohoto důvodu byl strom trénován pomocí náhodných snímků, které obsahovaly postavy jako je například řízení, tančení nebo běhání [9]. Klasifikátor pro přiřazení pixelu do dané části lidského těla pracuje na základě posouzení aktuálního pixelu a dvou dalších, které se nacházejí v různých odstupech od něj. Ten pak na základě hodnoty přiřadí daný pixel do segmentu za pomoci naučené pravděpodobnosti. Přiřazení jednotlivých pixelů do části lidského těla lze shlédnout na Obrázku 9. Více informací o této metodě lze získat v dokumentu [8] nebo v diplomové práci Ondřeje Černobila [9].



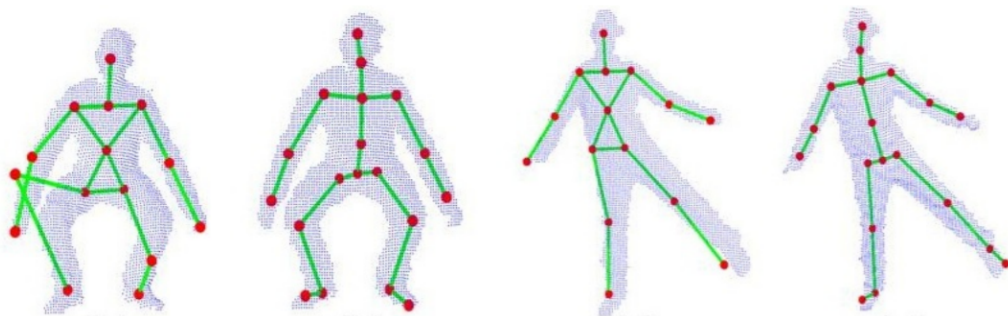
Zdroj: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf>

Obrázek 9: Znázornění jednotlivých částí lidského těla po přiřazení pixelů do dané oblasti.

3.3 Přesné odhady 3D póz z hloubkové mapy

Další metoda funguje taktéž na principu různě získaných póz, které jsou následně uloženy do databáze [10]. Jednotlivé pózy jsou získány za pomoci osmi-kamerového systému. V databázi je uloženo přibližně 19300 snímků, které obsahují chůzi, běh a ohýbání lidského těla. Tyto modely jsou doplněny o lidskou kostru, která obsahuje 19 spojení. Snímky lze shlédnout na Obrázku 10. Tyto snímky jsou následně pomocí různých animací rozšířeny o více pozic pro lepší porovnávání se získanými snímky z hloubkové mapy. Do vstupu pro porovnání je zahrnut jeden nebo více snímků získaných ze senzoru.

Postup, který autoři využívají je následující. Jako první je zapotřebí odstranit nežádoucí objekty z hloubkové mapy, kterými mohou být stoly, židle nebo objekty, které se vyskytují venku. To se provede na základě dvojitého prahování. Následně je zapotřebí ze snímku odstranit přebytečný šum. Jednotlivé snímky se převedou do kánonického koordinačního systému a následně se porovnávají se snímky v databázi jednotlivých póz. V databázi se nalezne snímek, který nejvíce odpovídá získanému obrysu lidského těla z hloubkové mapy. Kostra se poté optimalizuje přesně pro získaný snímek a zobrazí se kostra lidského těla. Detailnější informace o tomto algoritmu si lze přečíst v dokumentu autorů [10].



Zdroj: <https://www.inf.ethz.ch/personal/pomarc/pubs/YelICCV11.pdf>

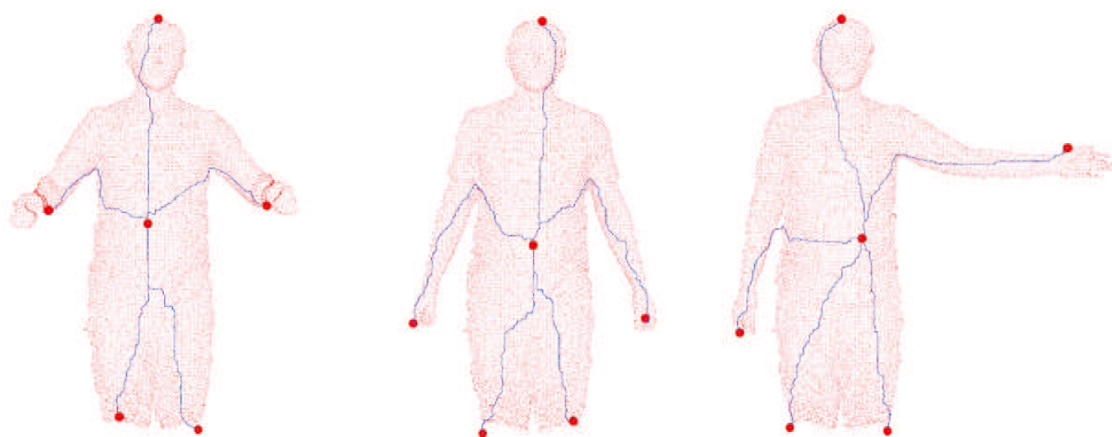
Obrázek 10: Znázornění snímků lidské kostry, které jsou uloženy v databázi pro následné porovnávání, včetně 19 spojení označených pomocí červených koleček

3.4 Tvorba 3D modelu řidiče na základě hloubkových map

Jako poslední vybrané existující řešení si popíšeme algoritmus, který je využit při tvorbě diplomové práce 3D modelu řidiče z hloubkové mapy. Jedná se o řešení navržené autorem Ondřejem Fuchsíkem a cílem je vytvořit kostru lidského těla [11]. Získání dat je docíleno za pomoci senzoru Kinect a následně se získaný snímek převede do mračna bodů za pomoci knihovny *Point Cloud Library*.

Jako první část je nalezena oblast trupu, ze které se následně budou hledat cesty do jednotlivých končetin a hlavy. Nejdříve se na základě hloubky odstraní pozadí a to tak, že uživatel zvolí hodnotu, za kterou již nebudou brány v potaz jednotlivé pixely obrazu. Část trupu se hledá na základě výpočtu vzdáleností mezi všemi body, které reprezentují části lidského těla, kterými jsou ruce, nohy a hlava. Vzdálenosti ke středu těla jsou ze všech bodů nejkratší, než ke kterémukoliv jiným bodům. Na základě tohoto lze identifikovat střed těla.

Jako další krok autor převádí mračno bodů za pomoci knihovny *Boost Graph Library*, aby bylo možné vytvořit z těchto bodů strukturu pro sestavení grafu. Po vytvoření grafu lze pak vyhledávat jednotlivé cesty. Pro vyhledávání jednotlivých částí končetin a hlavy je zde využito hledání cest s maximální délkou. Více informací o tomto postupu a detailnější popis lze vyčíst v diplomové práci autora [11]. Na Obrázku 11 lze shlédnout výsledek detekce lidské kostry.

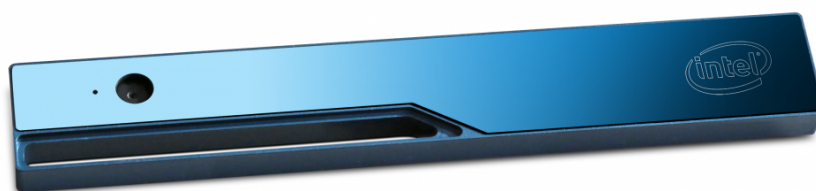


Zdroj: Tvorba 3D modelu řidiče na základě hloubkových map, FUCHSÍK Ondřej
<https://dspace.vsb.cz/handle/10084/115923>

Obrázek 11: Ukázka výsledků detekce lidské kostry na základě hloubkových map za pomoci navrženého algoritmu autorem diplomové práce.

4 Vlastní postupy a využití metody

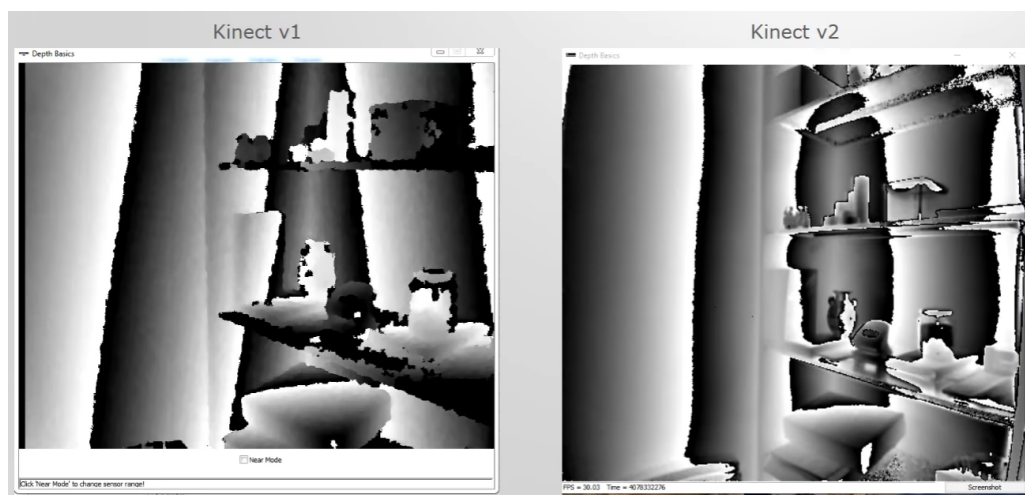
V této části se zaměříme na vlastní postup při realizaci práce. Na začátku této práce jsem se seznamoval s tím, jak vlastně Kinect funguje a jak lze z něj získat obraz. To, že dokážeme získat 3 typy obrazů jsme si popsali již dříve, ale pro nás byla nejdůležitější hloubková mapa. Hloubkovou mapu lze získat i za pomoci jiných senzorů. Ještě dříve, než jsem se hlouběji ponořil do tématu jsem uvažoval nad tím, jestli nelze získat hloubkovou mapu v lepší kvalitě než od Kinectu. Pro srovnání senzorů jsem využil také **Intel RealSense R200** (Obrázek 12), který také dokáže získat hloubkovou mapu, avšak po vyzkoušení jsem zjistil, že hloubková mapa není v takové kvalitě jakou požaduji, a proto jsem tento senzor vyloučil.



Zdroj: https://software.intel.com/sites/default/files/managed/b1/ce/R200_raw.png

Obrázek 12: Senzor Intel RealSense R200.

Lepší variantou však byl **Kinect v2**, který již disponoval lepší kvalitou získání hloubkové mapy. V obraze se nevyskytoval velký šum a rozsah obrazu byl větší než u varianty 1. Na Obrázku 13 lze vidět rozdíl mezi těmito verzemi. Lze vidět, že Kinect v2 má hladší hrany a v obraze se nevyskytuje zbytečný šum. Tento senzor však k dispozici nebyl, a tak jsem se rozhodl využít verzi 1, která mi byla zapůjčena.



Zdroj: <https://www.youtube.com/watch?v=FznorYFODCY>

Obrázek 13: Rozdíl získaného obrazu mezi Kinect v1 a Kinect v2.

4.1 Inicializace a získání dat z Kinectu

Abychom mohli pracovat s Kinectem, je zapotřebí jej nejprve nainicializovat. Pro tuto činnost využijeme vývojovou platformu Kinect SDK, která obsahuje několik API pro komunikaci s hardwarem, a tudíž i funkci pro inicializaci Kinectu [12]. Protože je Kinect schopen získávat obraz tří typů, je potřeba zvolit hloubkovou mapu. Dále nám Kinect umožňuje zvolit si z několika typů rozlišení a způsobů získání dat.

Po úspěšné inicializaci Kinectu můžeme následně přistoupit na získání jednotlivých snímků ze senzoru. Po získání jednoho snímku vypadá výsledný obraz jako na Obrázku 14.



Obrázek 14: Získání jednoho snímku z Kinectu.

Na obrázku lze vidět různé odstíny šedi, které značí jednotlivé vzdálenosti objektů od senzoru. Dále lze vidět, že hrany nejsou hladké, ale okousané. Dalším problémem je, že se hodnoty na určitých místech stále měnily, například z černé na bílou, vlivem senzoru. Nyní následuje otázka, jak se zbavit daného šumu a jak z obrazu dostat čistý obrys člověka bez jakéhokoliv šumu nebo jiných předmětů.

4.2 Odstranění pozadí a získání obrysu člověka

V této podkapitole si popíšeme postupy, které vedly k úspěšnému vyselektování člověka. První otázka byla, jak se vlastně pozadí zbavit. Jako první a základní metoda bylo takzvané **prahování** viz podkapitola 2.5. Jako druhou metodu jsem použil odstranění pozadí pomocí šablony, která bude popsána jako následující.

4.2.1 Prahování

Tato základní metoda byla využita jako prvotní myšlenka pro odstranění pozadí díky její jednoduché implementaci a rychlosti zpracování. Správné zvolení práhu lze docílit výpočtem histogramu, aby bylo dosaženo co nejlepšího prahování. Díky tomu lze odstranit pozadí a získat tak člověka. Bohužel čistě tato metoda není zcela funkční. Problém nastává již v Kinectu, který špatně získává hodnoty a občas je zaměňuje z 0 na 255 a naopak. To by však takový velký problém nebyl, ale další problém nastal v tom, že vzdálenost není dostatečná, aby bylo zcela jasně vidět rozdíly mezi člověkem a pozadím, jak můžeme vidět na Obrázku 15. V testovacím prostředí nemám možnost velkého prostoru, a tím se objevovaly nežádoucí objekty a člověk splýval se zdí, dveřmi a jinými objekty. Kvůli tomuto bylo potřeba vymyslet jinou metodu, která bude účinnější než tato. Jedinou výhodou této metody je to, že lze s Kinectem během snímání pohybovat a pozadí bude stále částečně odstraněno.



Obrázek 15: Porovnání vstupu a výsledku po prahování.

4.2.2 Odečet pozadí pomocí MOG, MOG2 a GMG

Nyní se podíváme na metody pro odečet pozadí. Odečet pozadí může být také znám jako detekce popředí. Tato technika se využívá ve zpracování obrazu a počítačovém vidění, například pro detekci různých objektů, jako jsou lidé nebo auta. Celá myšlenka spočívá v tom, že máme nějakou statickou kameru, kterou se nehýbe a tato kamera nám dokáže snímat rozdíly obrazů. Takže kamera má nějaké statické pozadí, které se nemění a v tom pozadí dokáže detekovat nově přichozí objekty. V OpenCV jsou pro tuto problematiku implementovány tři algoritmy, kterými jsou:

1. **BackgroundSubtractorMOG** - tato metoda je založena na modelování každého pixelu z obrazu na směs K Gaussových rozdělení ($K = 3$ až 5). Dále se zde nacházejí váhy, které reprezentují čas a značí, jak dlouho zůstávají v obraze pixely beze změny [17]. Z toho plyne, že jako barvy pozadí se jeví ty pixely, které jsou ve scéně déle a statické.

2. **BackgroundSubtractorMOG2** - tato metoda je podobná té předchozí. Zde akorát dochází k vylepšení a optimalizaci, kdy algoritmus je lépe adaptivní na změny světla ve scéně. V této metodě lze ještě zvolit, zdali chceme detekovat stíny nebo nikoliv. Stíny budou vyznačeny šedě, avšak sníží rychlost algoritmu [17].
3. **BackgroundSubtractorGMG** - tato metoda je kombinací statického obrazu pozadí a pixelové Bayesovy segmentace. Pro modelování pozadí se využívá prvních několika snímků, které poté umožňují detekci pozadí a objektů v popředí pomocí Bayesova rozhraní [17].

Na Obrázku 16 lze vidět konečné rozdíly mezi jednotlivými algoritmy.



Zdroj: http://docs.opencv.org/3.1.0/db/d5c/tutorial_py_bg_subtraction.html

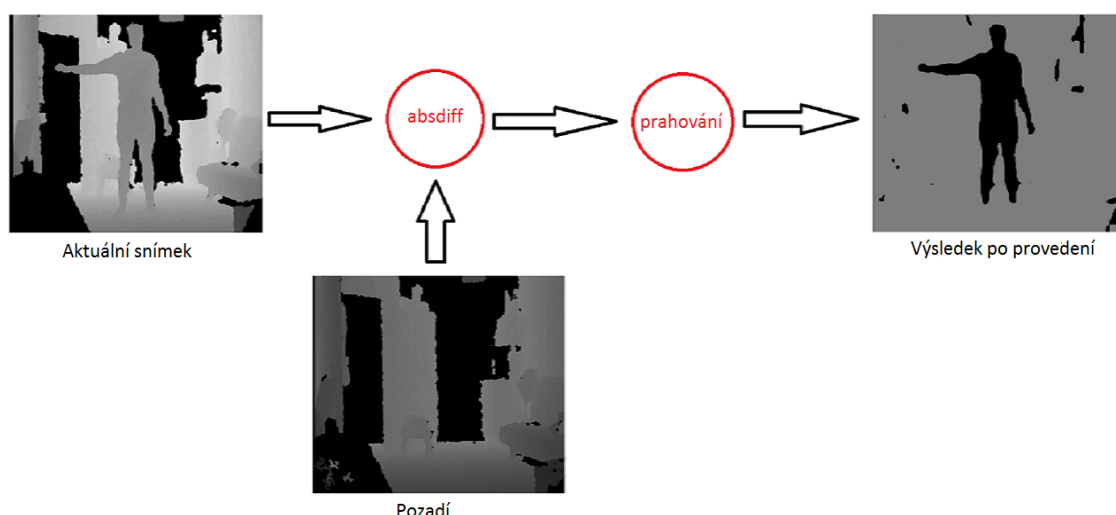
Obrázek 16: Rozdíl jednotlivých algoritmů pro odstranění pozadí.

Všechny tyto metody mají dobré využití a lze je využít i v implementaci pro rozpoznávání lidské kostry v hloubkové mapě. V programu je však využita ještě jedna metoda, která je jednodušší a pokusíme se nyní vysvětlit její funkčnost.

4.2.3 Metoda porovnání pozadí

První obrázek jako pozadí - takto by se dala nazvat funkce, která je vytvořená ve vlastní implementaci. Tato funkce využívá metodu s názvem **absdiff** [18], která je součástí OpenCV. Do této metody je zapotřebí poslat dva obrazy, přičemž jeden z nich je aktuální obraz z Kinectu a druhý je pozadí.

Pozadí získáme tak, že při každém spuštění programu se jako první vytvoří několik snímků z prostředí před kamerou. Tím získáme předlohu pro pozadí a můžeme ji využít pro naše účely. Získáme tedy aktuální snímek ze senzoru, oba dva obrazy vložíme jako parametry do funkce **absdiff** a výsledek uložíme do cílového obrazu. Pro vylepšení jsem využil i prahování, aby bylo docíleno co nejpřesnějšího výsledku. Při využití této metody je však zakázáno při real-time získávání obrazu jakkoliv hýbat se senzorem. V případě, že dojde k pohybu senzoru bude pozadí jiné a tato metoda nebude z důvodu odlišnosti mezi získaným pozadí při spuštění a aktuálním pozadí fungovat. Princip této metody lze vidět na Obrázku 17.



Obrázek 17: Princip metody pro odstranění pozadí, která je použita v programu.

Ještě předtím, než získáme výsledek aplikujeme na něj funkci **MedianBlur**. Jedná se o metodu, která pro daný pixel vypočítá střední hodnotu ze všech okolních pixelů v oblasti námi zvoleného [19]. Je to vysoce účinná metoda pro odstranění šumu v obrazech. Důvodem využití této metody je vyhlazení hran a vylepšení obrysu člověka pro další zpracování.

Když se pořádně podíváme na výsledek, lze vidět nežádoucí šum. Ten je způsoben tím, že Kinect nevydává data v dostatečné kvalitě a občas zaměňuje hodnoty jednotlivých pixelů z 0 na 255 a naopak, a proto se při odečtu pozadí i tak vyskytují nežádoucí objekty. Výše uvedené metody jsou pro tuto problematiku dostatečné, avšak ne zcela přesné. Představme si venkovní okolí, kde se mohou například vyskytovat stromy a jejich pohyb listů by mohl při těchto metodách dělat hodně velké problémy. Dále to mohou být také faktory, jako jsou počasí nebo ovlivnění osvětlení sluncem. V další podkapitole si popíšeme, jak lze odstranit jednotlivé nežádoucí objekty z výsledku po provedení metody.

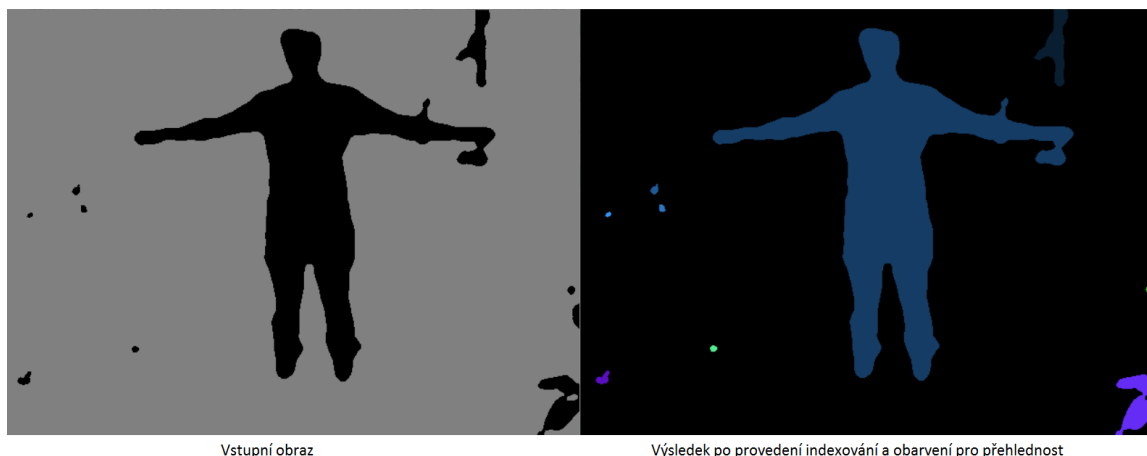
4.3 Odstranění nežádoucích objektů

Při pohledu na výsledek po provedení odstranění pozadí a následného prahování je potřeba se zamyslet nad tím, jak lze odstranit viditelné nežádoucí objekty, které jsou způsobeny vlivem nízké kvality snímání obrazu za pomoci senzoru, kdy se neustále dokola mění hodnoty na některých místech z 0 na 255 a naopak. Tímto pak vznikají nežádoucí objekty, kdy se získaný obraz neshoduje se získaným pozadím. Na začátku byla snaha o odstranění pomocí vylepšení prahování. Po dlouhých experimentech s hodnotou práhu nebylo dosaženo správného výsledku, a tak bylo potřeba vymyslet poněkud lepší algoritmus. Při pátrání po správné metodě přišel nápad, kdy bylo využito tzv. **Indexování**, které je popsáno v podkapitole 2.6.

4.3.1 Indexování

Abychom mohli pokračovat v detekci lidské kostry člověka v hloubkové mapě, bylo potřeba odstranit nežádoucí objekty. Pro tuto činnost se nejlépe zamlouvá indexování, které nám jednotlivé objekty očíslová.

Po správném zaindexování lze vidět výsledek, který je na Obrázku 18. Na obrázku lze vidět jednotlivé objekty, které jsou zaindexovány jiným číslem, a proto jsou obarveny jinak. Povšimněme si však chyby z našeho pohledu u pravé ruky, což je způsobeno malým prostorem pro testování, kdy ruka splývá s dveřmi. Během testování se však vyskytují střídavě i jiné objekty, které jsou způsobeny chybou zařízení Kinect.



Obrázek 18: Vstupní obraz pro zaindexování (vlevo) a výsledek po provedení (vpravo).

Nyní je potřeba nežádoucí objekty odstranit. To však nebude příliš obtížné, jelikož známe jednotlivé indexy všech objektů a dokážeme vypočítat obsah nebo obvod pro tyto objekty. Po získání hodnoty je myšlenka taková, že lze předpokládat, že největší obsah nebo obvod bude mít tvar člověka před naší kamerou.

4.3.2 Momenty, obvod a obsah

Nyní jsou již objekty od sebe úspěšně odděleny. Využijeme takové myšlenky, že každý objekt, který potřebujeme rozpoznat se skládá z číselných hodnot, tzv. příznaků. Tyto příznaky nesou o daném objektu podstatné informace, které lze rozlišit a na základě toho objekt identifikovat [15].

Jako první si popíšeme **momenty**. Ty patří k příznakům a jsou využívány docela často. Výhodou je, že jejich výpočet je velice snadný a většinou stačí pro rozlišení jednotlivých objektů. Momenty lze vypočítat dle předpisu $m_{p,q} = \sum_x \sum_y x^p y^q f(x,y)$ [20].

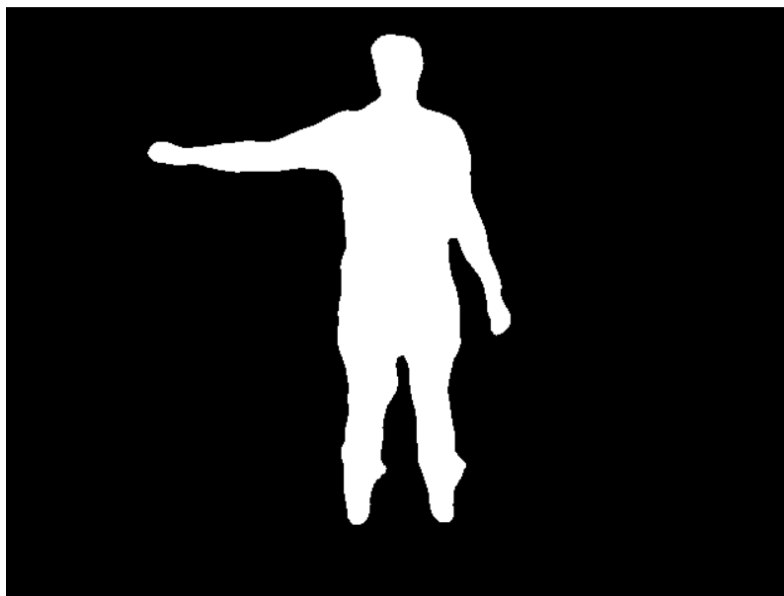
Nyní, když už víme jak vypočítat momenty, lze pomocí momentu $m_{0,0}$ vypočítat **obsah** objektu. Ten lze využít jako jeden z příznaků pro detekci našeho obrysu člověka. Po delším

testování vyšlo najevo, že největší obsah má právě člověk, a proto lze tento moment využít s velmi velkou přesností.

Druhým příznakem může být výpočet **obvodu**. Pro obvod je využit jednoduchý algoritmus, který pracuje na principu, kdy máme centrální pixel, a jestliže jeho čtyři sousední pixely obsahují stejné hodnoty, pak tento pixel není na obvodu daného objektu, a proto jej nelze započítat. V opačném případě je tento pixel započten jako pixel do obvodu.

V praxi to tedy funguje takto. Máme nějaké zaindexované objekty, kdy pro každý je jiná hodnota indexu, a tím jsme schopní je rozlišit. Pro každý objekt si vypočteme daný obvod pomocí algoritmu výše popsaného.

Nyní máme vypočtený obvod pro všechny objekty, a protože víme, že námi hledaný objekt má obvod největší, tak si jej přkopírujeme do nového čistého obrazu, a tím získáme čistý obrys člověka bez jakéhokoliv nežádoucího objektu. Výsledek lze vidět na Obrázku 19.



Obrázek 19: Výsledek po odstranění nežádoucích objektů pomocí indexování a nalezení objektu s největším obvodem.

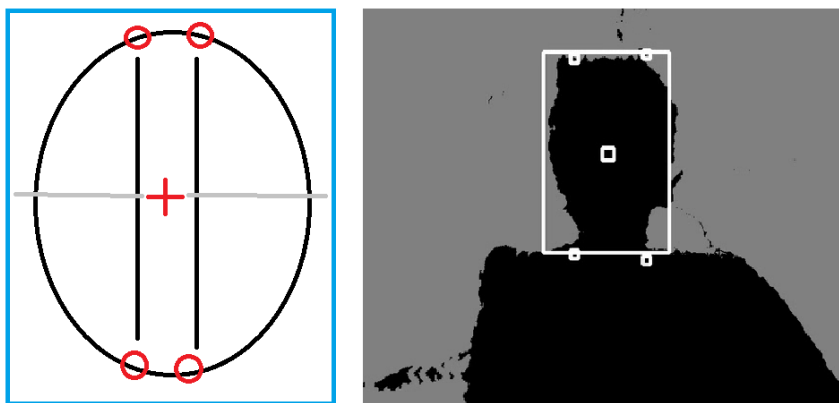
4.4 Nalezení hlavy

V této podkapitole si popíšeme jednotlivé metody, které vedly k nalezení hlavy. První metody, které byly vyvíjeny nebyly příliš úspěšné a byly převážně vyvíjeny při seznamování se s tématem. Nelze však vyloučit, že za určitých podmínek by tyto metody fungovaly, avšak by bylo třeba držet se striktně daných podmínek, kterých v reálném prostředí nemůžeme dosáhnout. První dvě základní metody byly vyvíjeny ještě dříve, než došlo k odstranění pozadí a jiných potřebných metod. Tyto metody byly využity za předpokladu, že hlava je hlavní součástí obrazu a je dostatečně v popředí. Dále následovaly poněkud složitější metody, které dosahovaly lepších výsledků.

4.4.1 Základní metoda č.1

Jedna z prvotních metod při vývoji byla metoda s názvem „**Dva body nahoře, dva body dole**“. Jedná se o zcela základní metodu, která byla vyvíjena na začátku, kdy se bylo třeba s tématem dostatečně seznámit, a tak její úspěšnost není příliš velká. V principu se jedná o způsob, který je znázorněn na Obrázku 20. Hlavní myšlenkou bylo mít hlavu rozdělenou na dvě poloviny. Dále bylo potřeba nalézt krajní body nahoře i dole. Tyto body byly nalezeny pomocí procházení jednotlivých řádků obrazu shora. Jakmile jsme narazili na černý pixel, bylo potřeba ověřit, zdali se opravdu jedná o hlavu, protože v počátečních experimentech nebylo odstraněno pozadí, a tak se mohlo jednat i o nežádoucí objekty. Zjistit, zdali se jedná o hlavu bylo v principu jednoduché. Bylo potřeba zjistit, zdali nalezený pixel pokračuje po řádcích dolů a na základě tohoto zjistit přibližnou velikost linie, ve které pokračuje. V případě, že se jednalo o samostatný pixel nebo nežádoucí malý objekt jsme tento pixel vynechali a pokračovali v nalezení dalšího bodu. Pomocí tohoto postupu jsme tak našli horní body. Dolní body byly omezeny po ramena a ty byly nalezeny na základě horních bodů, kdy jsme se posunuli po řádku do stran nalezených bodů. Dále jsme pokračovali po řádcích směrem dolů a jakmile jsme narazili na rameno, bylo možné určit dva body dole.

Na základě toho bylo možné získat přibližnou výšku hlavy a nyní bylo potřeba zjistit šířku, abychom mohli určit střed hlavy. Šířku stačilo dopočítat na základě poloviny výšky hlavy, kterou jsme dopočítali pomocí vzdáleností mezi nalezenými horními a dolními body. Na základě tohoto bylo možné zjistit střed mezi jednotlivými body a následně stačilo pouze sečíst jednotlivé pixely po obou stranách, které směřovaly ke kraji hlavy. Tato metoda byla vyvíjena s myšlenkou, že bude využita v místech, kde má hlava dostatečnou výšku a šířku jako je například ve vozidle, kde je kamera blízko řidiče. Výsledek lze vidět na Obrázku 20. Metoda však nemá stoprocentní úspěšnost z důvodu velice jednoduché logiky a při různém úhlu a naklonění hlavy docházelo k chybné detekci.



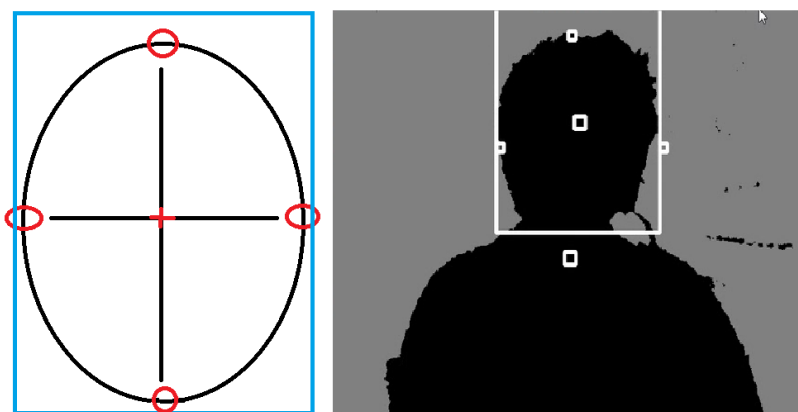
Obrázek 20: Metoda č.1 pro nalezení hlavy, která je založena na vyhledání dvou bodů nahoře a dvou bodů dole.

4.4.2 Základní metoda č.2

Tato metoda by se dala nazvat jako „**krajní body na každé straně**“. Je to další z metod, které se řadí mezi ty jednodušší a základní. Princip této metody je znázorněn na Obrázku 21. Jedná se o nalezení krajních bodů na každé straně, což je docíleno pomocí procházení obrazu ze všech směrů pomocí smyčky.

Pro nalezení horního bodu je potřeba procházet jednotlivé řádky obrazu, až narazíme na první černý pixel. V tomto případě je opět zapotřebí zjistit, zdali se opravdu jedná o hlavu a nejedná se o nežádoucí objekt z důvodu neodstraněného pozadí. Tohoto bylo docíleno pomocí stejného postupu, který je popsán v základní metodě č.1 v odstavci 4.4.1. Jakmile víme, že se opravdu může jednat o hlavu, můžeme pokračovat v nalezení krajních bodů. Ty jsou nalezeny pomocí procházení jednotlivých sloupců obrazu od shora dolů. Pro nalezení levého krajního bodu procházíme zleva doprava a pro nalezení pravého krajního bodu naopak. Při nalezení těchto bodů je stále zapotřebí kontrolovat, zdali se jedná o hlavu a v okolí tohoto bodu se nachází více černých pixelů, které mohou být součástí hlavy. Dolní bod lze nalézt na základě horního bodu a posunutí se po řádku do zvolené strany. Jakmile narazíme na rameno určíme přibližně spodní bod hlavy.

Nyní máme horní a spodní bod hlavy za pomoci kterých si vypočítáme přibližnou výšku hlavy na základě vzdálenosti mezi horním a spodním bodem, kdy od sebe odečteme pozici na řádku dolního bodu od pozice na řádku horního bodu. Tímto získáme přibližný střed hlavy, ale nyní je ještě zapotřebí vycentrovat krajní nalezené body. Pro oba krajní body známe jejich pozici a na základě nalezeného středu pomocí výšky hlavy vycentrujeme tyto krajní body na stejný řádek. Nyní nám chybí vypočítat pouze šířku hlavy a tu vypočteme stejným postupem jako výšku, kdy nám stačí nyní od sebe odečíst pozici na sloupci pravého krajního bodu od levého, a tím získáme přibližnou šířku hlavy. Na základě těchto hodnot lze určit střed hlavy, a tím nalezenou hlavu zvýraznit. Výsledek lze vidět na Obrázku 21. Tato metoda je však opět jednou ze základních, která byla vyvíjena zcela na počátku bez odstranění pozadí.



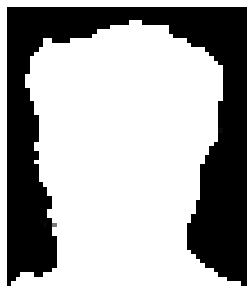
Obrázek 21: Metoda č.2 pro nalezení hlavy, která funguje na principu nalezení krajních bodů na všech stranách hlavy.

4.4.3 Detekce pomocí šablony (Template matching)

Nyní se podíváme na metodu, která je založena na porovnávání šablony se vstupním obrazem. Jelikož víme, jak hlava v hloubkové mapě vypadá, je možné ji nalézt pomocí šablony. Tato šablona bude fungovat ve většině případů, protože v hloubkové mapě nelze vidět rozdíly mezi jednotlivými obličejí, vlasy a podobně. Do výsledné matice se zapisuje pravděpodobnost, s jakou se šablona shoduje se vstupním obrazem. Kvůli tomu je zapotřebí dodržet jisté podmínky [21], které jsou:

- Předloha nesmí být větší než vstupní obraz.
- Výsledná matice musí mít přesně dané rozměry. Výsledná šířka musí mít velikost, která je dána šířkou vstupního obrazu, od kterého odečteme šířku šablony a přičteme jedničku. Stejným způsobem získáme rozměry pro výšku výsledného obrazu.
- Dále je potřeba mít výslednou matici datového typu float s bitovou hloubkou 32 bitů a jedním kanálem.

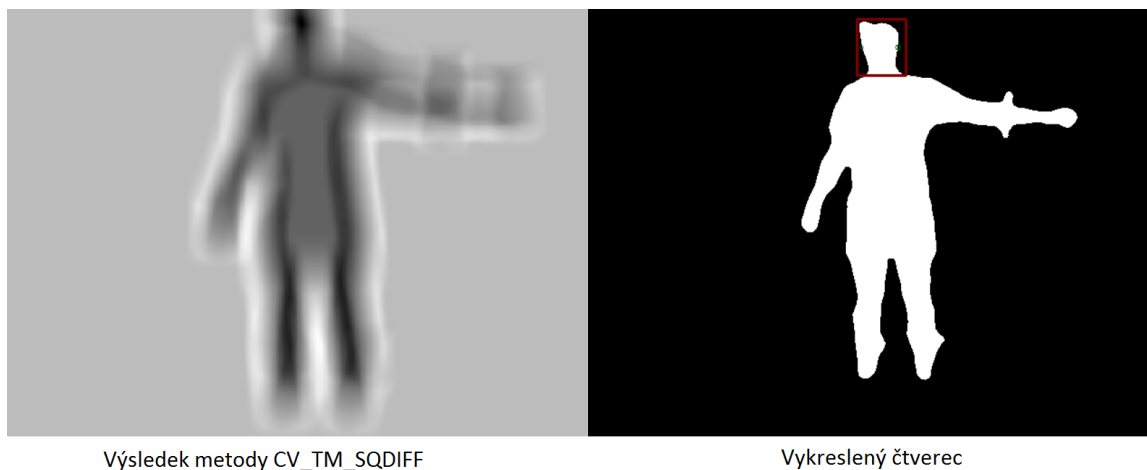
Nyní je potřeba si danou šablonu vytvořit. Jelikož máme podrobněji získaný obrys člověka v hloubkové mapě, není těžké si šablonu získat právě z tohoto obrysu. Vytvořenou šablonu lze vidět na Obrázku 22.



Obrázek 22: Šablona pro porovnání při detekci hlavy v obraze.

Dále je zapotřebí zavolat funkci **matchTemplate**, která je součástí OpenCV. Tato metoda umožňuje několik možností metod pro vytvoření výsledku do výsledné matice. Každá z těchto metod vrací jiné výsledky pro detekci šablony v obraze. Více informací a vzorce pro výpočet těchto metod lze zjistit v dokumentaci OpenCV v kapitole Template Matching [22]. V této diplomové práci je využita metoda s názvem **CV_TM_SQDIFF**, která vypočítává výslednou matici na základě předpisu $R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$, kde **T** značí šablonu a **I** značí vstupní obraz.

Výsledek z metody lze vidět na Obrázku 23 vlevo. Po získání výsledné matice je potřeba vyhledat hodnotu minima a maxima, a poté určit jejich pozice. Čtverec je pak vykreslen na pozici globálního extrému, který je znázorněn na Obrázku 23 vpravo.



Obrázek 23: Výsledek metody pro nalezení hlavy pomocí šablony (vlevo) a zobrazení nalezené hlavy pomocí červeného čtverce (vpravo).

Tato metoda však není zcela kvalitní. Nevýhodou je například to, když je objekt ze šablony v obraze jakkoliv zvětšený, zmenšený, otočený nebo zkosený. Předností této metody je vyhledávání objektů v barevných obrazech. V barevném obraze je tato metoda účinná, avšak obraz nesmí být nijak deformován. Po různých testech vyšlo najevo, že lze tuto metodu využít i v hloubkové mapě, avšak za předpokladu, že velikost hlavy bude stále stejná. To v našem případě nemůže být nikdy stoprocentně zaručeno, protože v reálných podmínkách se nám při různém pohybu může velikost hlavy měnit. Nevýhodu této metody lze eliminovat přípravou několika šablon, které budou obsahovat různé velikosti hlavy. Jelikož se pohybujeme v hloubkové mapě, nemusíme brát v úvahu různé typy obličejů a můžeme se spolehnout na to, že lze tuto metodu využít v jakémkoliv případě a na jakoukoliv osobu. V tomto případě není nijak výsledek ovlivňován intenzitou světla, zatímco v barevném obraze by různé odstíny a přisvícení mohlo dělat problém.

4.5 Nalezení ramen

V případě, že je hlava úspěšně a správně nalezena, následuje další část implementace, která slouží k nalezení bodů, které značí ramena. Protože předem víme, že velikost ramen je s největší pravděpodobností větší než šířka hlavy, tak se z tohoto důvodu budeme od ní odrážet. Nejkratnější body ramen lze odvodit přibližně od dvojnásobku šířky hlavy.

Nejprve si tedy vypočítáme šířku hlavy. Protože jsme v této práci využili vyhledání hlavy pomocí šablony, známe zde pouze střed hlavy. Vypočítat však šířku není nijak složitá funkce. Jedná se pouze o výpočet všech pixelů v rovnoběžné linii středu hlavy. Pro přesnější šířku je výhodnější výpočet provést po obou stranách od středu, protože střed hlavy nemusí být přesně detekován uprostřed, ale detekovaný čtverec může být vlivem špatné detekce mírně posunutý. Při výpočtu šířky hlavy si současně určíme krajní body hlavy v rovnoběžné linii, které jsou znázorněny na Obrázku 24 vpravo zelenými kolečky. Jak jsem již výše zmínil, ramena mají

přibližně dvojnásobnou šířku hlavy, a proto je lze detekovat tak, že ke krajním bodům hlavy připočteme posun přibližně o jednu čtvrtinu šířky hlavy do strany, a tím se dostaneme přibližně na střed ramen, která jsou znázorněna červenými kolečky. Algoritmus tohoto řešení lze vidět na Obrázku 24 vlevo.



Obrázek 24: Nalezení ramen lidského těla v hloubkové mapě za pomoci krajních bodů hlavy, posunutí do stran od těchto bodů a následného posunutí směrem dolů, až narazíme na ramena.

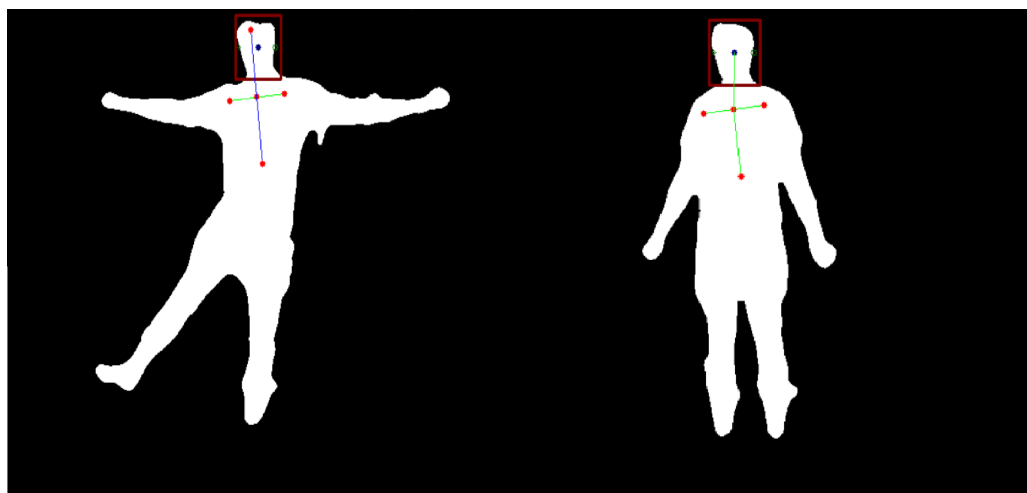
Po nalezení dvou krajních bodů pro ramena je potřeba vypočíst ještě střed ramen, který bude potřebný pro následující detekci středu těla a spodní části, která bude použita k detekci kyčlí. Střed získáme pomocí vzdálenosti mezi krajními body ramen. Tuto vzdálenost získáme pomocí odečtení pozice ve sloupci levého bodu od pravého. Po vydělení této vzdálenosti dvěma získáme střed. Tuto metodu lze dobře využít v případě, že je člověk otočený přímo ke kameře. V případě, že by byl člověk otočený ke kameře bokem, metoda nebude schopná správně detekovat ramena, a tím může být ovlivněná celá kostra lidského těla, protože na ramenech závisí ruce i tělo a následná detekce kyčlí. Výsledek tohoto algoritmu lze vidět na Obrázku 24 vpravo, včetně propojení jednotlivých bodů pro lepší viditelnost nalezené kostry.

4.6 Nalezení kyčlí

Po nalezení ramen a jejich středu přichází na řadu detekce dalších částí lidského těla a těmi jsou kyčle. Abychom se ke kyčlím dostali, je nejprve potřeba detekovat střed těla a následně se dostat ke spodní části trupu.

Pro nalezení střední části těla využijeme výpočet normály od středu ramen. Představme si přímku mezi dvěma body detekovaných ramen jako rovinu. V tomto případě je normála přímka, která je kolmá na danou rovinu, v našem případě na přímku. Rovina je dvourozměrný geometrický útvar, který si lze představit jako dokonale rovnou plochu [23]. Normálu lze vypočítat na základě rovnic $dx = x_2 - x_1$, $dy = y_2 - y_1$, $normal1 = (-dy, dx)$ a $normal2 = (dy, -dx)$. Hodnota x_1 značí pozici levého ramene na ose x a hodnota y_1 značí hodnotu levého ramene na

ose y . To samé platí i pro index 2, který značí hodnoty pravého ramene. Hodnota *normal1* značí normálu ze středu ramen ve směru dolů a hodnota *normal2* značí normálu ve směru nahoru. Vypočtené normály lze vidět na Obrázku 25 vlevo, které jsou zvýrazněny modrou barvou. Pro správné určení trupu je zapotřebí opět stát tělem přímo na kameru. Protože je výpočet normály závislý na detekci ramen, na kterých je závislá detekce trupu, je nutné, aby ramena byla detekována správně. V případě, že by ramena byla nalezena špatně, odvíjela by se špatně i detekce těla a následně kyčlí. Nalezený střed trupu lze vidět na Obrázku 25 vpravo.

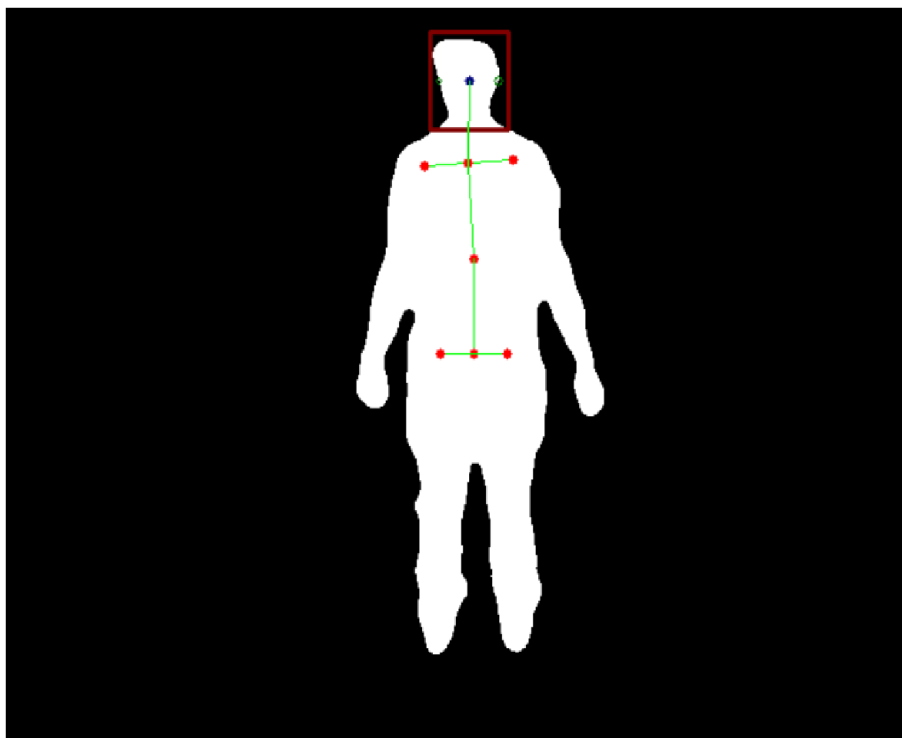


Obrázek 25: Znázornění vypočtených normál ze středu ramen.

Nyní jsme našli první bod směřující ke spodní části trupu. Abychom mohli nalézt kyčle je zapotřebí vypočítat normálu ještě jednou za pomoci stejného postupu, avšak nyní z bodu, který značí střed trupu. Důvod pro vypočtení normál je takový, abychom dosáhli co nejpřesnějšího středu ve spodní části těla. Jakmile známe střed v oblasti kyčlí, je zapotřebí tyto kyčle nalézt.

Nalezení kyčlí je dosaženo pomocí vypočtení šířky trupu. Šířku trupu lze vypočítat na základě sečtení bílých pixelů na řádku nalezeného bodu ve spodní části. Sečtení pixelů je potřeba provést po obou stranách z důvodu následného určení kyčlí. Jakmile známe šířku trupu, můžeme určit jednotlivé body. To provedeme tak, že k nalezenému středu dolní části trupu připočteme čtvrtinu šířky trupu. Tím získáme nový bod, který označuje pravou část kyčle. Pro nalezení levé kyčle pouze odečteme čtvrtinu šířky trupu. Tímto postupem získáme oba body, které označují kyčle. Na Obrázku 26 lze vidět oba zobrazené body včetně jejich spojení zelenou přímkou, která označuje kostru lidského těla.

Nyní se téměř blížíme ke konečnému výsledku, který bude obsahovat nalezení lidské kostry v hloubkové mapě včetně končetin. V další podkapitole se podíváme na nalezení horní části končetin, jimiž jsou lokty a ruce.



Obrázek 26: Výsledné nalezení kyčlí ve spodní části trupu za pomoci výpočtů normál ze středu ramen směrem dolů.

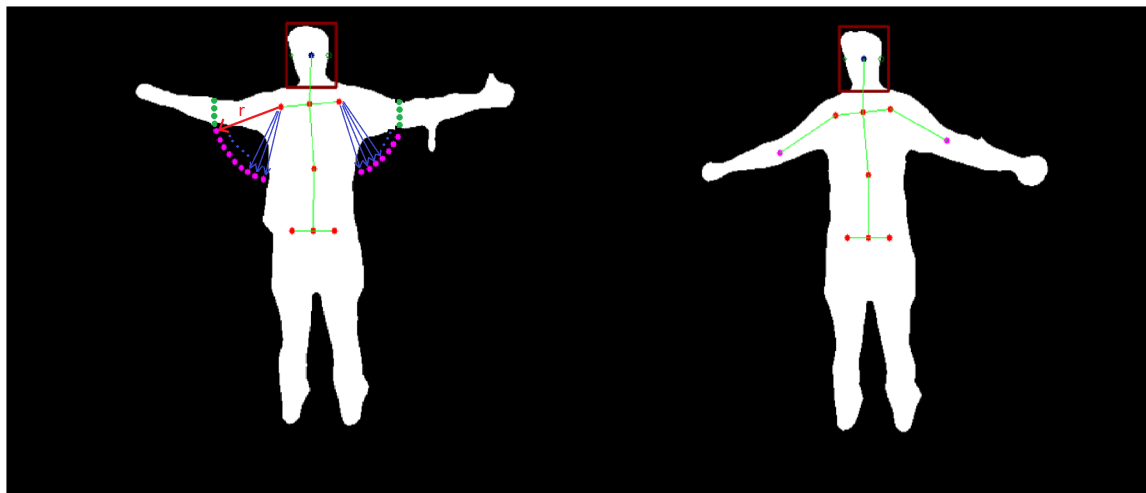
4.7 Nalezení loktů a rukou

Po nalezení kyčlí ve spodní části trupu nás v této podkapitole čeká nalezení loktů a rukou. Pro detekci loktů je zapotřebí znát body obou ramen, která jsme detekovali v podkapitole 4.5.

Tato metoda je založena na vyhledávání bodů na kružnici s daným poloměrem a určení bodu, který se nachází v oblasti lokte a následného aplikování stejného postupu v oblasti lokte pro nalezení rukou. Jako první je zapotřebí si určit poloměr, ve kterém budeme vyhledávat část lokte. Ze stavby lidského těla lze přibližně určit vzdálenost mezi ramenem a loktem jako vzdálenost mezi středem ramen a středem trupu. Poloměr tedy vypočítáme tak, že od pozice řádku středu trupu odečteme hodnotu pozice středu ramen na řádku. Protože není přesně určeno, kde se loket může nacházet je zapotřebí v případě, že se loket nenajde, v daném poloměru jej zmenšit o určitou vzdálenost. V našem případě se v programu zmenšujeme o hodnotu 5. Zmenšování poloměru o zvolenou hodnotu se provádí po každém procházení bodů na kružnici v případě, že loket nebude nalezen.

Výpočet bodů na kružnici, na kterých má být detekován loket lze vypočíst na základě rovnic $Bod_x = x + r * \cos(a * \pi / 180)$ a $Bod_y = y + r * \sin(a * \pi / 180)$. Bod nám značí pozici jednotlivých bodů na kružnici. Hodnoty x a y značí pozici bodu levého nebo pravého ramene, poloměr je označen písmenem r a úhel je označen písmenem a . V případě, že se na daném bodě nenachází pixel patřící ruce, pokračuje se ve výpočtu dalšího bodu v daném poloměru s posunutým úhlem

a v případě, že se již všechny body vyčerpají snížíme poloměr. Princip nalezení lokte na kružnici lze vidět na Obrázku 27 vlevo. Fialové body znázorňují jednotlivé body na kružnici, na kterých se detekuje, zdali se nachází v oblasti ruky. Zelené body značí detekované body, které se nachází v oblasti lokte a uloží se do vektoru, ze kterého se pak následně vybírá prvek nejbližší středu.

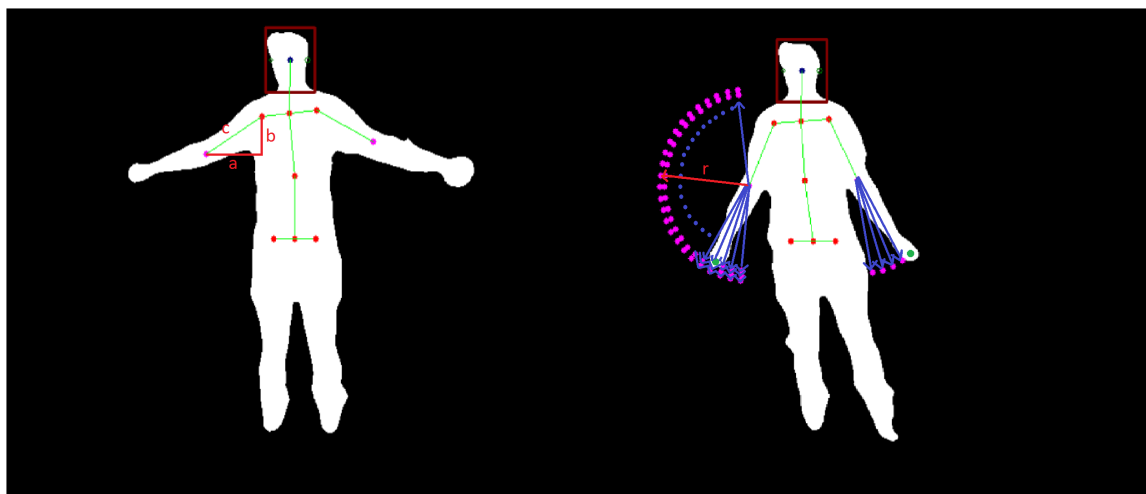


Obrázek 27: Princip nalezení loktů za pomoci bodů na kružnici (vlevo) a zobrazení výsledné detekce loktů (vpravo).

V případě, že se nalezený bod nachází v části ruky je také nejprve potřeba ověřit, zdali je přímka mezi ramenem a loktem bez průchodu mimo ruku. Lze předpokládat, že mezi ramenem a loktem nemůže být něco jiného než ruka, a proto je zapotřebí projít všechny body na přímce mezi těmito částmi těla. V případě, že se na přímce nenachází žádný černý pixel lze předpokládat, že se mezi těmito body nenachází žádné přerušení, a proto můžeme označit nalezený bod jako loket. Pokud je však přímka přerušovaná a není souvislá po oblasti ruky, tento bod vynecháme a pokračujeme ve stejném postupu nalezení, který je popsán výše. Pro vylepšení detekce loktů se všechny body, které se vyskytují v oblasti rukou uloží do vektoru. Následně se z vektoru vybere ten prvek, který se nachází nejbližší středu lokte, a tím je většinou prvek, který se nachází uprostřed mezi všemi prvky vektoru. Aby byly lokty správně nalezeny je zapotřebí, aby se ruce nenacházely před tělem, protože by v tomto případě ruka splývala s tělem vlivem nízké kvality získávání obrazu ze senzoru. Pokud by k tomuto případu došlo a po maximálním snížení poloměru by nebyl loket nalezen nastane případ, kdy se nebude pokračovat dále ve vyhledávání ruky a detekce ruky bude ukončena. Dále může nastat případ, kdy se v oblasti lokte bude nacházet nežádoucí objekt způsobený například dveřmi, které budou mít stejnou vzdálenost od senzoru jako ruce. V tomto případě je využita kontrola, kdy mezi ramenem a loktem musí být nepřerušovaná přímka. Jelikož nám objekt může zvětšit šířku ruky v oblasti lokte, je pro tento případ použit výše zmíněný vektor se všemi možnými body, kdy se vybere ten prostřední, a tím je docílena správná detekce lokte. Výsledek nalezení loktů lze vidět na Obrázku 27 vpravo.

Nyní máme nalezené lokty a je zapotřebí nalézt ještě ruce. Tento postup bude téměř totožný

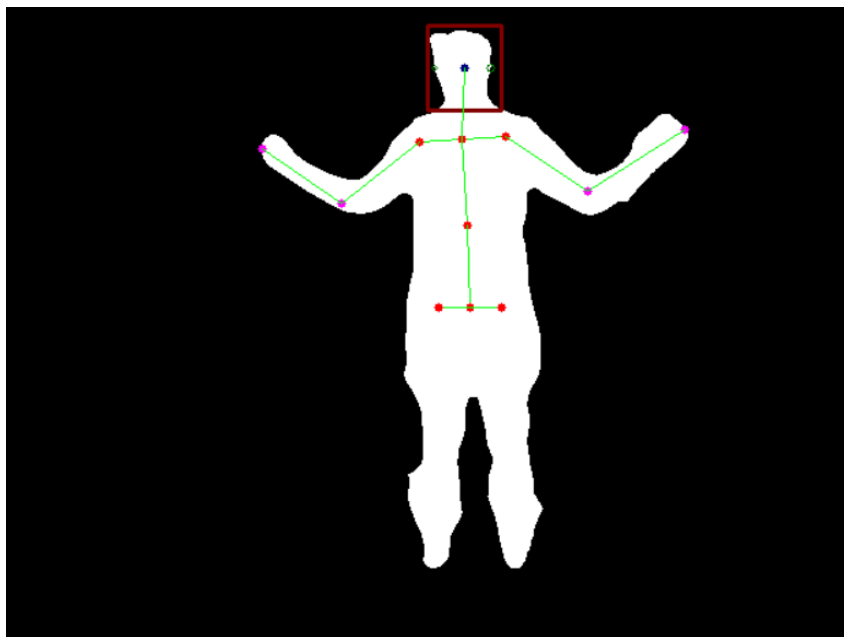
s tím, který jsme využili při nalezení loktů. Zde však je zapotřebí vypočítat poloměr, ve kterém bude vyhledávaná ruka. Poloměr je zapotřebí vypočítat pomocí Pythagorovy věty, abychom měli přesnou vzdálenost mezi bodem ramene a loktem. Protože známe pozice obou bodů, mezi kterými potřebujeme vypočítat vzdálenost, nebude to pro nás problém. Vypočítat délku lze na základě rovnice $c^2 = \sqrt{a^2 + b^2}$. Hodnota c značí délku, kterou chceme zjistit. Hodnota a je vzdálenost mezi body levého lokte a ramene ve sloupcích. Hodnota b je vzdálenost mezi řádky. Znázornění lze vidět na Obrázku 28 vlevo.



Obrázek 28: Znázornění výpočtu vzdálenosti mezi ramenem a loktem pomocí Pythagorovy věty (vlevo) a princip nalezení rukou pomocí bodů na kružnici (vpravo).

Nyní víme poloměr, ve kterém je zapotřebí detekovat ruce. Princip detekce rukou probíhá stejně jako detekce loktů. Nyní se však odražíme od lokte, ze kterého procházíme body na kružnici v daném poloměru a zjišťujeme, zdali se bod nachází v oblasti ruky. Po nalezení bodu je potřeba opět kontrolovat, zdali přímka mezi těmito dvěma body není přerušená a v případě, že by byla, je zapotřebí nalézt jiný bod. Pokud se v daném poloměru nenalezne žádný bod, který by mohl patřit do oblasti ruky, je zapotřebí snížit poloměr a opakovat postup. Pro vyhledávání ruky platí stejné podmínky jako pro nalezení lokte. Ruce se nesmí vyskytovat před tělem nebo se přímo dotýkat nohy. V tomto případě nebude ruka správně nalezena a může dojít k záměně ruky například za stehno viz Obrázek 37. Princip hledání bodů lze vidět na Obrázku 28 vpravo. Fialové body značí procházené body a zelené body značí nalezené, které patří do oblasti ruky. Červeně je vyznačen námi vypočtený poloměr.

Po nalezení všech bodů jsme schopni zobrazit kostru lidského těla včetně rukou, a nyní nás čeká detekce kolen a nohou. Výsledek lze vidět na Obrázku 29 včetně spojení jednotlivých bodů pro lepší znázornění a viditelnost. Tato metoda funguje pouze na vyhledávání jednotlivých bodů na daném poloměru kružnice a vzhledem k její nenáročnosti na implementaci, funguje celkem spolehlivě. Pro dosažení úspěšnosti je zapotřebí, aby byly dodrženy výše uvedené podmínky jako jsou přímý postoj ke kameře, ruce nesmí být před tělem nebo se nesmí dotýkat nohou.



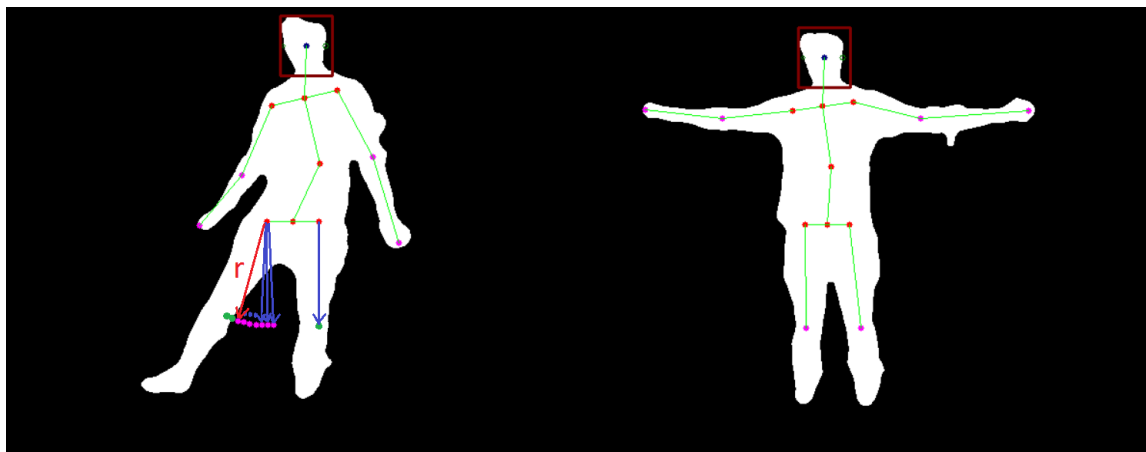
Obrázek 29: Výsledná detekce rukou za pomoci procházení bodů na kružnici.

4.8 Nalezení kolen a nohou

Nyní se blížíme ke konečnému výsledku a zbývá nám pouze nalézt nohy. Abychom mohli detekovat nohy je nejprve zapotřebí nalézt kolena, která budeme detekovat za pomoci kyčlí, které jsme úspěšně detekovali v podkapitole 4.6.

Pro nalezení kolen využijeme podobný princip, který byl použit při detekci rukou. Jedná se tedy o metodu, kdy procházíme jednotlivé body na kružnici s daným poloměrem a zjišťujeme, zdali se daný bod nachází v oblasti kolen. Velikost poloměru lze vypočítat na základě vzdálenosti mezi nalezeným bodem středu těla a středem mezi kyčlemi a to tak, že od pozice na řádku bodu mezi kyčlemi odečteme pozici na řádku od bodu, který značí střed těla. Po různých experimentech bylo zjištěno, že lze k této vzdálenosti přičíst jednu třetinu navíc, abychom dosáhli lepší detekce kolene.

Pro výpočet bodů na kružnici lze opět využít rovnice $Bod_x = x + r * \cos(a * \pi/180)$ a $Bod_y = y + r * \sin(a * \pi/180)$. Jakmile tyto body známe, je potřeba zjistit, zdali se kontrolovaný bod nachází v oblasti kolene. To lze zjistit tak, že nalezený pixel na dané pozici musí být bílý. Způsob, kterým jsou detekovány kolena lze vidět na Obrázku 30 vlevo. Na obrázku se nacházejí opět fialové body, které značí procházené body a dále zelené body, které značí body v oblasti kolene. Jakmile zjistíme, že se daný bod nachází v oblasti kolene tak je nejprve zapotřebí opět zjistit, zdali přímka mezi kyčli a nalezeným bodem není přerušena. Tuto podmínku zajistíme stejným způsobem jako u rukou a to tak, že projdeme všechny body na přímce a v případě, že žádný pixel není černý lze tento bod označit jako koleno. Pokud je však tato přímka přerušena je zapotřebí nalézt jiný bod na kružnici.



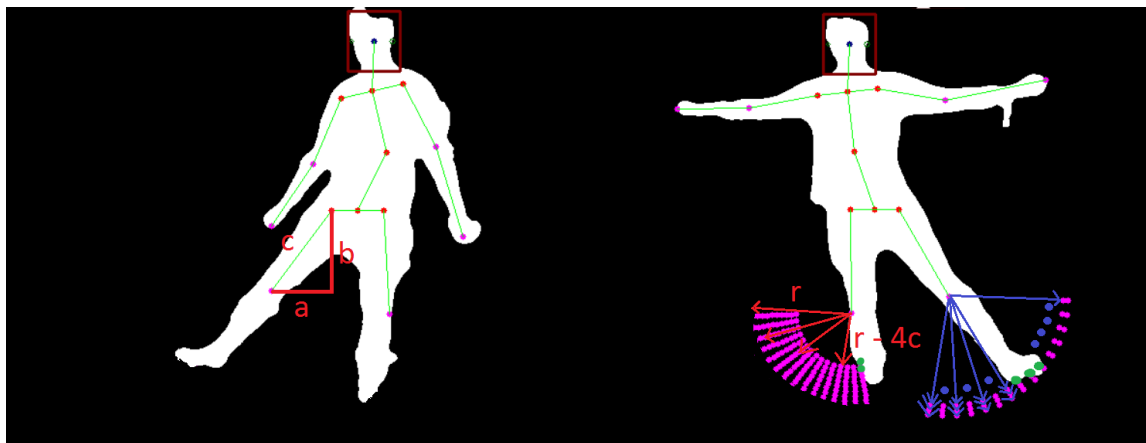
Obrázek 30: Princip nalezení kolen s využitím bodů na kružnici (vlevo) a výsledná detekce kolen za pomoci této metody (vpravo).

Problémy mohou nastat například v případě, kdy je člověk opět otočený ke kameře bokem. Dalším důležitým faktorem je, aby byly správně detekovány kyčle, na kterých je detekce kolen postavena. Další podmínkou pro úspěšnou detekci je například to, že je zakázáno jakkoliv nohy překřížovat. V případě překřížení nohou dojde k chybné detekci z důvodu nerozeznání jednotlivých oblastí, kdy nevíme, která část patří ke které noze, vlivem nízké schopnosti rozeznání vzdálenosti nohou od senzoru. Kvůli tomu jsou vzdálenosti od sebe nerozpoznatelné a program se tváří, že se jedná o jednu a tu samou nohu. Pro správnou funkčnost detekce nohou je tedy zapotřebí mít nohy jasně rozeznatelné. V případě nežádoucího objektu v oblasti kolene může dojít k mírné odchylce ze středu označení. V případě, že by nebylo koleno nalezeno je následná detekce chodidla ukončena. Jak byla kolena nalezena lze shlédnout na Obrázku 30 vpravo.

Po úspěšném nalezení kolen se blížíme do finální fáze, a to je detekce chodidel. Postup pro nalezení je opět podobný jako při nalezení kolen nebo rukou v předchozím případě. Poloměr pro kružnici, na které budeme vyhledávat se vypočítá za pomoci rovnice $c^2 = \sqrt{a^2 + b^2}$. Jedná se opět o Pythagorovu větu a v našem případě je nyní hodnota a jako vzdálenost mezi kolenem a kyčlí ve sloupcích a hodnota b je vzdálenost mezi těmito body na řádcích. Pro lepší představu lze tento princip vidět na Obrázku 31 vlevo. Po výpočtu poloměru lze stejným postupem nalézt chodidla. Procházíme jednotlivé body na kružnici a zjišťujeme, zdali se nenachází v oblasti, které by mohly patřit chodidlům. Při detekci je opět potřeba zjišťovat, zdali přímka mezi nalezeným bodem a kolenem není přerušena. Jak nalézt bod pro chodidlo lze vidět pro lepší představu na Obrázku 31 vpravo, kdy jsou opět procházené body označeny fialově a body, které se nachází v oblasti chodidla jsou označeny zeleně. Na obrázku lze také vidět označení poloměru r a označení $r-4c$, kde c značí konstantu, o kterou zmenšíme poloměr v případě, že v předchozím poloměru nebyl nalezen žádný bod.

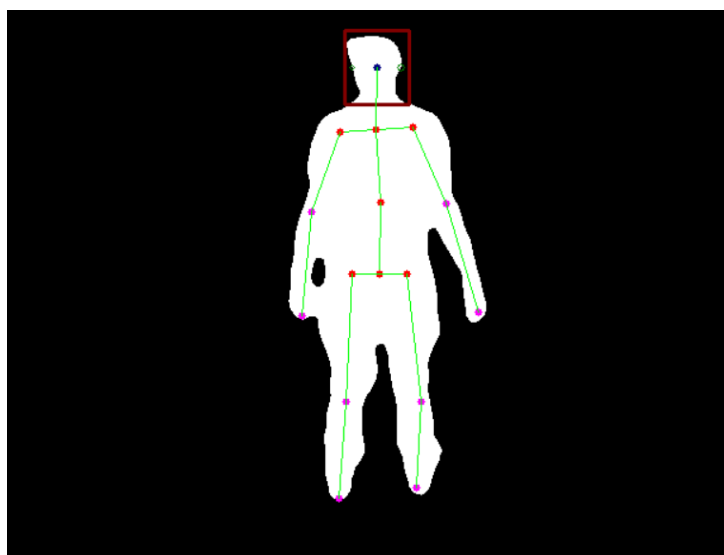
Problém při detekci chodidel může nastat v případě, že jsou kolena špatně detekována. Tato metoda také neuvažuje případ, kdy si člověk může před kameru kleknout. V tomto případě by

nohy byly detekovány na stejné úrovni jako kolena. I zde platí podmínka, kdy se nohy nesmí překřížovat ani nijak dotýkat s rukou. Na Obrázku 31 vpravo lze také shlédnout, že levá spodní část chodidla není vidět. Je to způsobeno tím, že senzor snímá spodní část nohy zároveň s podlahou a vlivem nízkého rozlišení vzdálenosti od senzoru se část nohy počítá jako podlaha a je při odstranění pozadí smazána.



Obrázek 31: Znázornění výpočtu vzdálenosti mezi kyčlí a kolenem za pomoci Pythagorovy věty(vlevo) a princip nalezení chodidel při procházení bodů (vpravo).

Nyní již máme všechny potřebné body k tomu, abychom úspěšně byli schopni detekovat nohy. Výsledek včetně propojení jednotlivých bodů lze vidět na Obrázku 32. Nohy byly poslední detekce, o kterou v této práci šlo a nyní již máme detekovanou celou lidskou kostru těla v hloubkové mapě. V další kapitole se podíváme na experimentální výsledky a úspěšnost při detekci.



Obrázek 32: Výsledek nalezení nohou za použití algoritmu detekce bodů na kružnici.

5 Experimentální část

V této části se podíváme na experimenty s různými pozicemi člověka a otestujeme tak úspěšnost námi navržených metod pro detekci lidské kostry v hloubkové mapě. Podíváme se na nepřesnosti, chyby v detekci a popíšeme si příčiny těchto chyb. Dále si zobrazíme tabulku s procentuální úspěšnosti, která bude zjišťována na různých datových sadách.

5.1 Datová sada

Testování probíhalo na přístroji s dvoujádrovým procesorem Intel Core i5-4210H s frekvencí 2.90GHz, operační paměti 8GB RAM a grafické kartě nVidia 940M. Při této sestavě se rychlost programu pohybovala kolem 22 snímků za vteřinu, což je zcela dostačující pro plynulý chod programu. Abychom mohli řádně otestovat navržený algoritmus pro detekci je zapotřebí mít dostatečnou testovací sadu s různými pozicemi lidského těla před senzorem. Program je taktéž schopen pracovat real-time, avšak pro lepší orientaci a nalezení chyb je zapotřebí si jednotlivé snímky ukládat pro pozdější kontrolu.

První datovou sadou se jeví snímky pořízené na začátku tvorby této práce. Tyto snímky byly pořízeny pomocí senzoru a byly využity při realizaci práce. Lze tedy říci, že na základě tohoto videa byly vyvíjeny všechny algoritmy a optimalizace probíhala právě na tomto videu. Ukázka je přiložena na CD ve složce **VIDEOA**, včetně výsledného videa detekce. Jedná se o video s názvem **Video1.avi**.

Druhým videem je datová sada, která obsahuje jednotlivé pozice lidského těla, kterými jsou:

- Základní postoj s rukama podél těla.
- Základní postoj s rozpaženýma rukama.
- Základní postoj s jednou rukou rozpaženou a druhou podél těla.
- Ruce za hlavou.
- Jedna z noh a ruka jdou do strany.
- Ruce před tělem.

Jednotlivé postoje jsou detekovány a vyhodnocovány na základě pořízených snímků z detekce. Veškeré snímky jsou pořízeny, kdy je člověk otočený přímo na kameru. Ukázku postojů, které budou testovány lze vidět na Obrázku 33 bez jakýchkoliv úprav, které probíhají v této práci pro úspěšnou detekci lidské kostry. Video je přiloženo pod názvem **Video2.avi**.

Jako třetí použijeme náhodně natočené video různých pozic a pohybů lidského těla, které bylo vytvořené na konci vývoje této práce. Toto video se nachází ve složce pod názvem **Video3.avi**. V následující podkapitole si zobrazíme výsledek detekce pro základní postoje a probereme jednotlivé chyby, které nastaly při detekci ve všech videích.



(a) Ruce podél těla.

(b) Rozpažené ruce.



(c) Jedna ruka rozpažená.

(d) Ruce za hlavou.

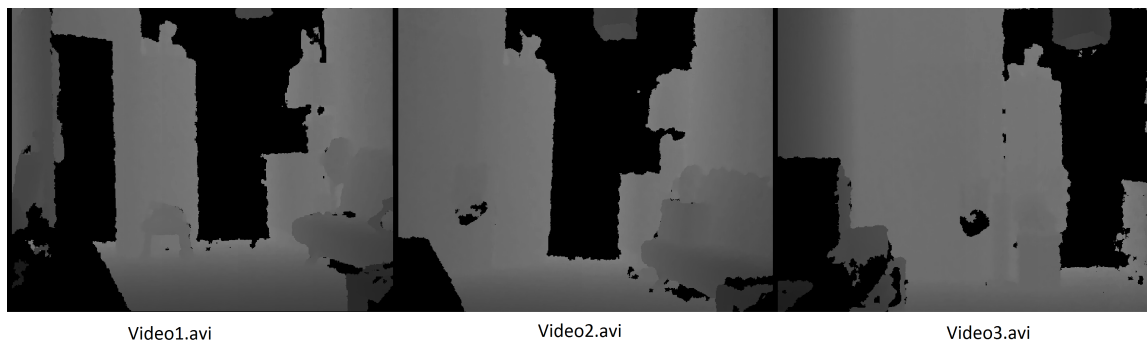


(e) Noha a ruka do strany.

(f) Ruce před tělem.

Obrázek 33: Ukázka jednotlivých postojů, které jsou použity pro detekci lidské kostry v hloubkové mapě.

Všechna výše uvedená videa byla natáčena za stejných podmínek. Z důvodu nedostatku místa pro optimální snímání obrazu je možné ve všech videích vidět okolí jako je například pohovka, lednice nebo židle. Všechny tyto objekty ztěžují práci při odstranění pozadí a komplikují správné rozlišení vzdálenosti od člověka k senzoru tím, že mohou mít téměř stejnou vzdálenost, a tím jsou pak přidruženy k člověku a brány jako jeden objekt. Samozřejmě jsme se těchto objektů zbavili pomocí metody odstranění pozadí, avšak při pohybu ruky může docházet k tomu, že jsou k rukám přidruženy například dveře, které lze vidět na Obrázku 23 vpravo, kdy se v oblasti pravého lokte vyskytuje část dveří. Jako druhý problém může nastat například to, že vlivem malého prostoru se ruka ztratí a splyne s dveřmi, a tím je brána ruka jako pozadí a bude odstraněna. Přehled jednotlivých pozadí, které jsme využili ve všech třech videích lze vidět na Obrázku 34. Povšimněme si, že se na obrázku vyskytují i černé oblasti. Ty se zde vyskytují vlivem omezení rozsahu detekce vzdálenosti za pomoci senzoru, který je schopen snímat do vzdálenosti tří až čtyř metrů. Jelikož místnost dále pokračuje, senzor není schopen vyslat paprsek, který by byl schopen tuto vzdálenost detekovat, přechází jej a převést na hloubkovou informaci.



Obrázek 34: Přehled pozadí, které se vyskytovaly před senzorem během snímání a natáčení videí.

5.2 Výsledky detekce

V této podkapitole si ukážeme jednotlivé detekce vzorových postav a podíváme se na chyby v detekci v různých datových sadách. Dále si vypočítáme procentuální úspěšnost detekce lidské kostry v hloubkové mapě.

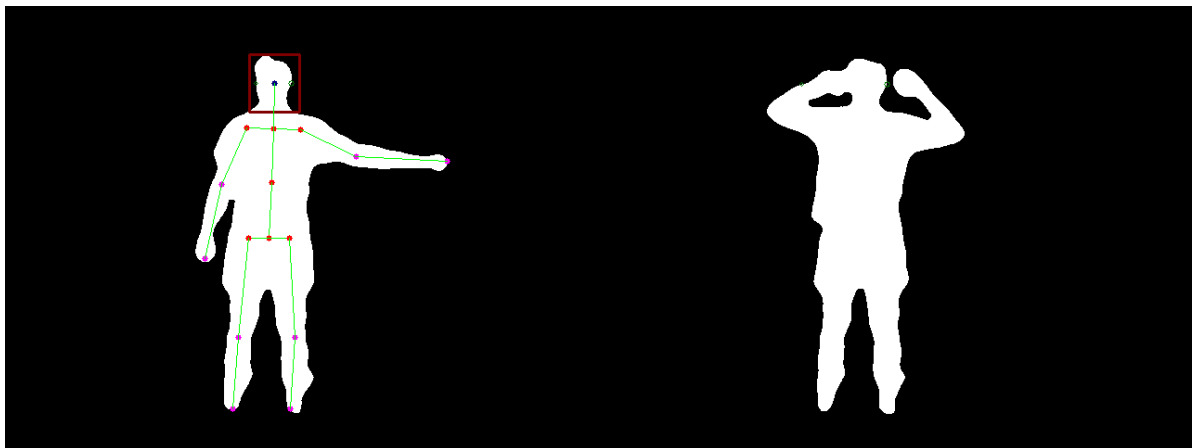
5.2.1 Hraniční případy

Nyní se podíváme na hraniční případy, kdy nastávají chyby při detekci lidské kostry. Popíšeme si důvody těchto chyb a případná vylepšení, která by mohla tyto chyby odstranit. Na Obrázku 35 lze vidět výsledné detekce postav. Úspěšná detekce včetně rukou a nohou proběhla na čtyřech ze šesti obrázků.



(a) Ruce podél těla.

(b) Rozpažené ruce.



(c) Jedna ruka rozpažená.

(d) Ruce za hlavou.



(e) Noha a ruka do strany.

(f) Ruce před tělem.

Obrázek 35: Výsledek detekce zvolených postojů po použití navrženého postupu.

5.2.1.1 Ruce za hlavou

Problém na Obrázku 35 část (d), kde jsou ruce za hlavou spočívá v tom, že hlava je vyhledávaná pomocí šablony. Šablona je pevně daná a lze ji shlédnout na Obrázku 22. Důvod, proč není hlava detekována je ten, že se ruka dotýká hlavy, a tím je porušena shoda mezi šablonou a vstupním obrazem. Abychom mohli hlavu správně detekovat je zapotřebí mít co nejpřesnější shodu mezi šablonou a obrazem. Velikost hlavy musí mít přibližnou velikost jako je v šabloně a nesmí být nijak natočená, protože metoda pro vyhledávání pomocí šablony je schopná detekovat pouze přesně podle šablony. Kostra lidského těla není zobrazena, protože již dříve jsme popsali, že se detekce odvíjí od hlavy a v případě, že hlava není nalezena se detekce ukončí.

5.2.1.2 Ruce před tělem

Jako další problém se jeví ruce před tělem. Výsledek detekce lze shlédnout na Obrázku 35 část (f). Problémem u této detekce je to, že Kinect verze 1 není schopen kvalitně rozlišit vzdálenost mezi senzorem, tělem a rukama. Vzdálenosti jsou mírně odlišné, avšak rozdíl mezi zobrazením jednotlivých vzdáleností je minimální a rozlišení tak rukou od těla je velmi obtížné již na pohled. Z toho důvodu jsem se rozhodl tuto situaci vynechat a brát tak v potaz, že se ruce při detekci nesmějí nacházet před tělem. Pro vylepšení této detekce by bylo vhodné vyzkoušet získání hloubkové mapy pomocí senzoru Kinect verze 2, která má vylepšené rozlišení a kvalitu snímání obrazu.

5.2.1.3 Špatné natočení těla k senzoru

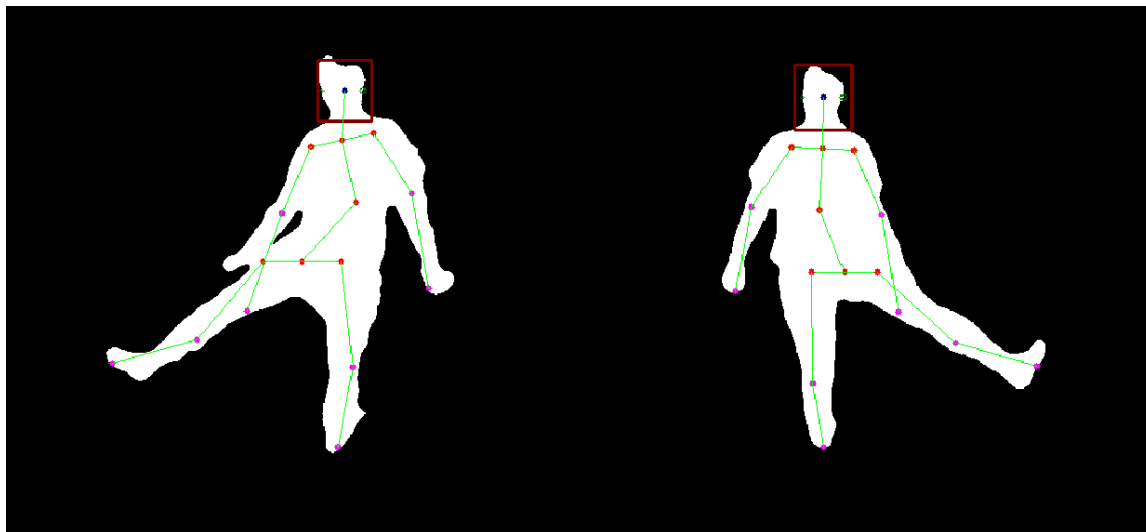
Dalším problémem při detekci lidského těla je natočení těla ke kameře. Algoritmus je schopen detekovat správně pouze v případě, že je lidské tělo natočeno přímo na kameru senzoru. V případě, že je tělo otočeno jiným směrem dochází k chybné detekci, kterou lze shlédnout na Obrázku 36. Lze vidět, že hlava je úspěšně detekována, avšak problém nastává již při detekci rukou nebo například při výpočtu normály pro detekci středu těla.



Obrázek 36: Problémy při špatném natočení těla k senzoru.

5.2.1.4 Ruka se dotýká nohy

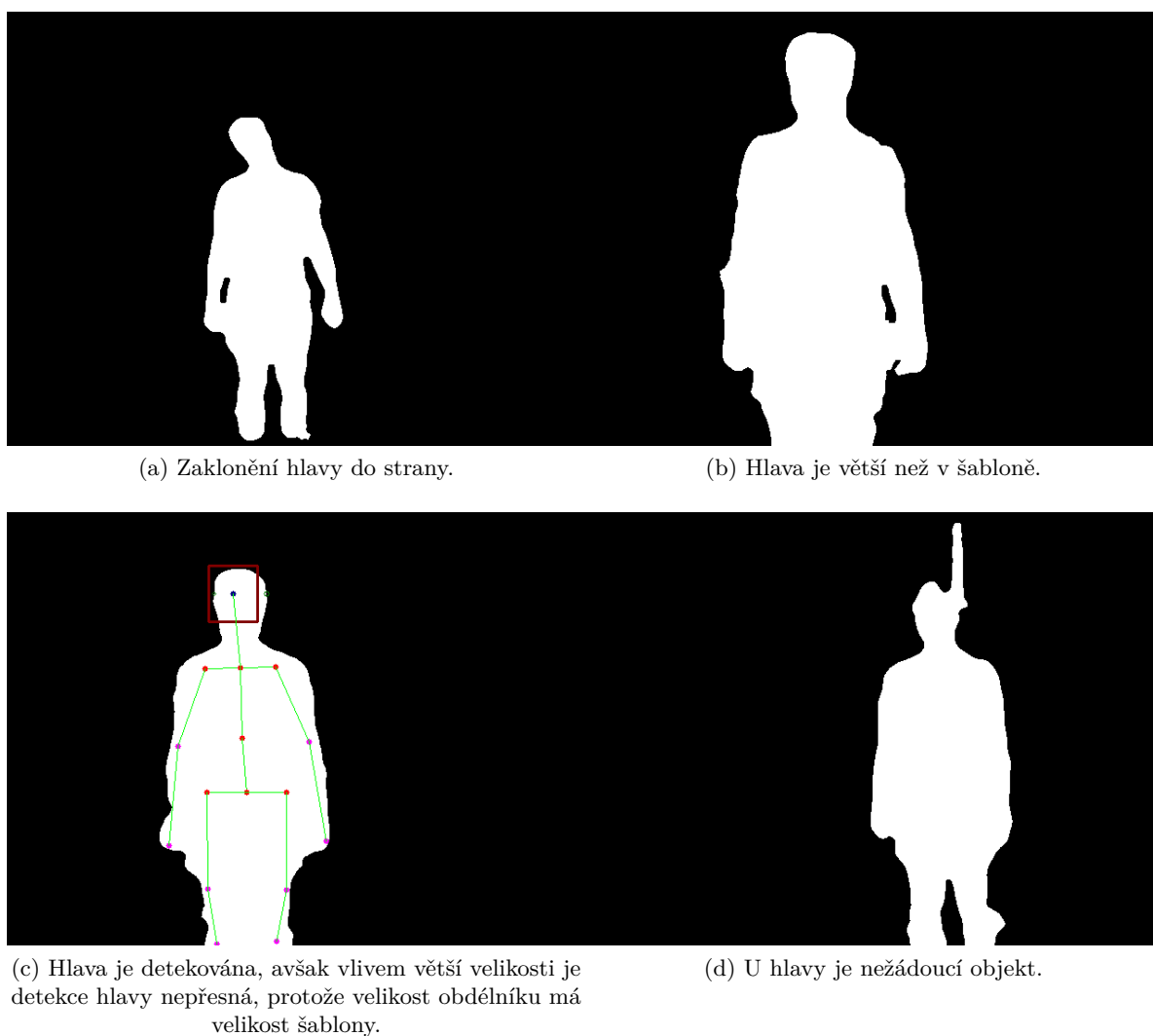
Dalším hraničním případem u detekce je dotýkání se rukou nohy v takovém rozsahu, že je délka hranice spojení mezi rukou a nohou dostatečně velká na to, aby se bez přerušení spojily body mezi rukou a loktem. Protože nedojde k přerušení pomocí černého pixelu, který značí okolí mimo tělo, detektor si myslí, že je nalezený bod součástí ruky a označí jej. Tento problém lze shlédnout na Obrázku 37, kde jde vidět jak je detekována ruka na pozici stehna, protože se na lince mezi loktem a detekovaným bodem nenachází žádný černý pixel. V případě oddělení ruky od nohy je opět ruka detekována správně.



Obrázek 37: Problémy při doteku ruky a nohy, které nastávají při detekci.

5.3 Špatné natočení hlavy

Dalším problémem je špatné natočení hlavy, který jsme si již dříve popsali. Nyní se podíváme na to, jak se tento problém projevuje v praxi. V případě chybné detekce hlavy dojde k ukončení jakékoliv detekce lidské kostry. Protože je hlava nejdůležitější část v našem programu je zapotřebí, aby tato část byla detekována úspěšně. Pro úspěšnost detekce je však potřeba se držet jasně daných pravidel, kdy hlava nesmí být nijak natočená, zvětšená a podobně. Kolem hlavy se nesmí objevovat žádné nežádoucí objekty vlivem špatné kvality získávání hloubkové mapy. Jako nežádoucí objekt můžeme brát například světlo, které se nachází blízko hlavy vlivem malého testovacího prostoru. Dále by mohl nastávat problém v případě, že bude mít člověk na hlavě čepici s bambulkou či jiný neobvyklý tvar, který by vadil při detekci a hlava by nebyla shodná se šablonou. Na Obrázku 38 lze shlédnout jednotlivé chybné detekce lidské hlavy v hloubkové mapě.



Obrázek 38: Chybné detekce hlavy vlivem nedodržení podmínek pro úspěšnou detekci pomocí šablony.

5.4 Vyhodnocení úspěšnosti

Následuje podkapitola, ve které si vypočítáme procentuální úspěšnost detekce na základě tří sekvencí snímků (tři videa). Úspěšnost bude rozdělena na dvě tabulky. První část je detekce hlavy jako samotná. Na základě získaných snímků, ze kterých je následně vytvořené video se při průchodu těchto snímků bude zjišťovat, zdali je hlava úspěšně detekována. Druhá tabulka se bude věnovat úspěšnosti detekce lidské kostry jako celek. Při průchodu snímků se bude kontrolovat detekce všech lidských částí, o kterých lze říci, že jsou detekovány správně nebo s minimální odchylkou.

Pro testování jsme tedy využili 3 různě natočená videa, která obsahují mírně odlišné pozadí, jiné pohyby a odlišné pózy lidského těla. Kontrola úspěšnosti probíhá od snímku, kde se poprvé

nachází rozumný obrys člověka a nejsou brány v potaz nepodstatné snímky, abychom dosáhli co nejpřesnější úspěšnosti. První video je tvořeno přibližně **1450** snímky, druhé video obsahuje **1037** snímků a poslední třetí video má **735** snímků. Na základě počtu snímků se pak bude vypočítávat úspěšnost detekce. Video byla natáčena nezávisle na sobě, a tak by jejich úspěšnost měla být rozdílná. Při natáčení se bral ohled na velikost hlavy, která je nejdůležitějším faktorem pro celou detekci. Všechna tři videa jsou natáčena za předpokladu, že jsou dodrženy všechny podmínky pro detekci hlavy a tělo bylo nasměrováno přímo k senzoru.



Obrázek 39: Příklad špatné a správné detekce hlavy ve Videu 1.

V Tabulce 1 lze shlédnout úspěšnost detekce hlavy na jednotlivých videích. Jak jsme si již dříve popsali, detekce hlavy je závislá na velikosti a natočení hlavy směrem k senzoru. Na všech třech videích se úspěšnost detekce pohybuje kolem 97 procent. Chyby při detekci také nastaly v případech, že se kolem hlavy objevil nežádoucí objekt spojený s hlavou nebo se ruka dotýkala hlavy. Při tomto problému se detekce nepovedla, protože hlava nebyla stejná jako šablona vytvořená pro detekci. Příklad správné a špatné detekce hlavy lze vidět na Obrázku 39, která byla prováděná na Videu 1. Vlevo lze vidět chybnou detekci hlavy vlivem ruky, která se dotýká hlavy a vpravo je správně detekována hlava včetně následující detekce kostry.

Tabulka 1: Úspěšnost algoritmu při detekci hlavy.

	Správná detekce	Špatná detekce	Celková úspěšnost
Video 1	1410	40	97,24%
Video 2	1012	20	97,58%
Video 3	717	18	97,55%

V Tabulce 2 lze vidět úspěšnost pro detekci lidské kostry jako celek. Snímky jsou kontrolovány s určitou odchylkou, kdy lze ještě detekci částí považovat za úspěšnou. V případě, že se již detekce liší významně, kdy kostra ztrácí smysl, je zaznamenána za chybnou. Chyby se vyskytují například při doteku hlavy rukou, když se dají ruce před tělo nebo ruka splývá s nohou, jak jsme si již řekli v odstavci 5.2.1.

Tabulka 2: Úspěšnost detekce lidského těla při použití algoritmu.

	Správná detekce	Špatná detekce	Celková úspěšnost
Video 1	1325	125	91,37%
Video 2	837	200	80,71%
Video 3	643	92	87,48%

Na Obrázku 40 je vlevo ukázaná správná detekce lidské kostry a vpravo se nachází detekce chybná. Chyba je způsobena velkým náklonem těla do strany, kdy je výpočet úhlů odvíjen od ramene. Počáteční úhel pro procházení je nastaven tak, aby se při přímém postoji začalo procházet těsně od boku, aby se povedla detekce ruky podél těla. Možné vylepšení této metody je takové, že lze vypočítat náklon těla a na základě tohoto posunout počátek detekce ruky o námi vypočtený úhel. Pravá ruka je detekována na pozici stehna, protože první nalezený bod ve svislé poloze od ramene je právě na noze, a proto se již nedostaneme ke správné detekci ruky. Stejná chyba je způsobena u detekce levé ruky, kdy je loket detekován těsně na kraji, avšak při detekci ruky, již nastává chyba vlivem právě nakloněného těla.



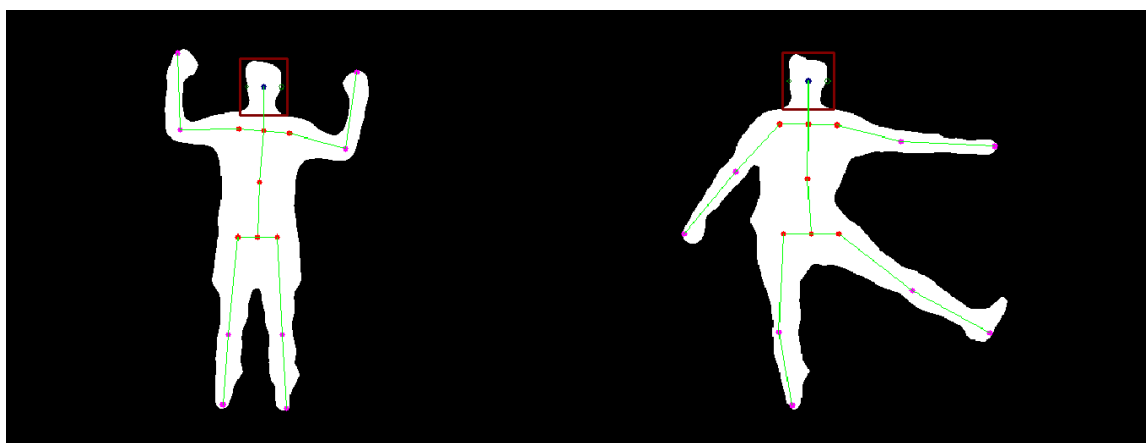
Obrázek 40: Příklad správné a špatné detekce celé lidské kostry ve Videu 3.

5.5 Možná vylepšení

Nyní se podíváme na možnosti vylepšení, které by mohly této práci pomoci. Protože nikdy nedosáhneme optimálních podmínek pro detekci lidské kostry podíváme se aspoň na části, které by mohly vylepšit algoritmus, který je navržený v této práci.

- **Kinect verze 2** - jako základní prvek pro vylepšení se jeví Kinect v2. Na Obrázku 13 jsme si porovnali rozdíly mezi jednotlivými kinecty, přičemž novější kinect disponuje lepším senzorem, a tudíž je schopen lépe rozlišovat vzdálenosti mezi objekty detailněji a bylo by možné detekovat rozdíly, například když jsou ruce před tělem.

- **Větší prostor** - vývoj této práce probíhal v domácím prostředí, kde byla kvalita obrazu ovlivňována prostředím. Větší prostor pro pohyb člověka značí kvalitnější hloubkovou mapu bez objektů, které by splývaly s tělem lidského těla a poté byl obrys člověka špatný. Při testování se stávalo, že při nedostatku prostoru člověk splýval se zdí nebo dveřmi, a tím nebylo možné úspěšně získat obrys člověka.
- **Detekce hlavy** - hlava je v této práci nejdůležitějším prvkem, od kterého se odráží veškerá detekce. Metoda použita v této práci má dobrou úspěšnost, avšak je zapotřebí se držet striktně daných podmínek jako je velikost hlavy, její natočení a nesmí obsahovat spojené nežádoucí objekty. To je však v reálných podmínkách těžko dosažitelné. Pro vylepšení detekce hlavy by bylo možné využít metodu pomocí **Histogramů orientovaných gradientů (HOG)** [24] [25]. Základem této metody je myšlenka, že vzhled a tvar objektu je charakterizován pomocí intenzity gradientů nebo směru hran. Obraz je rozdělen do několika částí, ve kterých je sestaven histogram gradientů. Protože lze předpokládat, že hlava bude mít intenzitu gradientů většinou stejnou a směr hran bude podobný mohla by tato metoda vylepšit detekci hlavy a nahradit tak současnou metodu. Následně lze využít SVM klasifikátor pro detekci hlavy [26].
- **Detekce zbývajících částí těla** - detekce ostatních částí těla závisí na kvalitě získané hloubkové mapy, kdy by se překrývání částí dalo pomocí lepšího rozlišení vzdálenosti lépe odlišit. Dále by se daly vylepšit již navržené algoritmy jako je například výpočet bodů na kružnici na přesnější nalezení bodů, které se nachází na částech těla. Při natočení těla se zřejmě optimalizace již nedá vymyslet, protože v případě, že bychom viděli pouze bok těla nemá smysl detekovat jednotlivé části, které by ani nebyly rozeznatelné. Na Obrázku 41 lze vidět úspěšnou detekci lidské kostry ve Videu 2, která je nalezena právě za pomoci algoritmu procházení bodů na kružnici.



Obrázek 41: Znázornění úspěšných detekcí lidské kostry pořízené ve Videu 2.

6 Závěr

V této práci bylo cílem vytvořit postup, který měl za úkol detekovat lidskou kostru v hloubkové mapě. Jako snímací zařízení jsme využili senzor zvaný Kinect verze 1, který však má nižší kvalitu získávání obrazu než Kinect verze 2. Pomocí senzoru jsme získali hloubkovou mapu, kterou bylo potřeba několika způsoby upravit a následně použít vytvořené algoritmy pro správné určení lidské kostry.

Na začátku dokumentu jsme se podívali na základní informace a metody, které jsou v této práci využity. Práce je vytvořena v programovacím jazyku C++ za pomoci využití volně dostupné knihovny OpenCV. Také jsme se podívali na již existující postupy, které vedou k úspěšné detekci hlavy, která je pro naši práci velmi důležitá, a také na postupy, které vedou k úspěšné detekci lidské kostry v hloubkové mapě.

Dále jsme si řekli vlastní navržený postup pro detekci. Po získání hloubkové mapy ze senzoru bylo zapotřebí odstranit pozadí, abychom mohli získat přesný obrys člověka. Odstranění pozadí bylo provedeno za pomoci metody, kdy jsme si vytvořili počáteční snímky pozadí, a ty následně porovnali s aktuálním snímkem a odstranili. Po odstranění pozadí se zde vyskytovaly nežádoucí objekty, které byly odstraněny pomocí indexování obrazu a následného vypočítání obvodu, kdy objekt s největším obvodem je právě námi hledaný obrys těla. Poté jsme pokračovali k nalezení hlavy, kdy jsme použili metodu detekce pomocí šablony. Tato metoda měla dostatečnou úspěšnost, avšak pouze za dodržení pevně daných podmínek, které jsme si popsali. V závěru bylo zapotřebí nalézt jednotlivé končetiny. Ruce jsme detekovali na základě nalezených ramen, od kterých jsme použili metodu procházení bodů na kružnici pro nalezení loktů i dlaní. Detekce nohou probíhala v podobném principu, kdy jsme pomocí této metody detekovali dolní končetiny od nalezených kyčlí.

V experimentální části jsme si ověřili úspěšnost navržených metod. Úspěšnost detekce hlavy se pohybovala nad 90 procentní úspěšnosti. Detekce hlavy je však omezena podmínkami, které jsme si dříve popsali. Dále jsme testovali detekci lidské kostry jako celek na základě několika pevně daných postojů, a dále za pomoci natočených videí, která obsahovala různé pohyby a pozice těla. Úspěšnost se pohybovala od 80 do 90 procent. Problémy nastávaly například při velkém naklonění lidského těla do strany nebo se ruce nesměly dotýkat hlavy popřípadě jiných končetin. Dále také nastávala chyba při umístění rukou před tělo, kdy vlivem senzoru nebylo možné dostatečně rozlišit vzdálenost rukou a těla od senzoru nebo v případě, že tělo nebylo natočeno přímo na kameru.

Výsledkem této práce je úspěšná detekce lidské kostry v hloubkové mapě. Práce je splněna ve všech bodech, které byly zadány. Využití tohoto algoritmu si lze představit například ve vozidlech jako kontrolu, zdali má řidič ruce na volant a nespí. Na přiloženém CD se nachází jednotlivá videa s výslednou detekcí, programátorská část, elektronická podoba této práce, spustitelná část práce bez nutnosti instalace vývojového prostředí a návod, jak tuto práci spustit.

Literatura

- [1] C++. *Wikipedia.org* [online]. poslední aktualizace 2017-02-09 [cit. 2017-02-10].
Dostupné z: <https://en.wikipedia.org/wiki/C%2B%2B>
- [2] TIOBE Software. *Tiobe Index* [online]. únor 2017 [cit. 2017-02-10].
Dostupné z: <http://www.tiobe.com/tiobe-index/>
- [3] Kinect. *Wikipedia.org* [online]. poslední aktualizace 2017-01-19 [cit. 2017-02-12].
Dostupné z: https://en.wikipedia.org/wiki/Kinect#Kinect_for_Windows
- [4] Kinect for Windows Sensor Components and Specifications. *Microsoft* [online]. [cit. 2017-02-12]. Dostupné z: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>
- [5] About OpenCV. *opencv.org* [online]. [cit. 2017-02-12].
Dostupné z: <http://opencv.org/about.html>
- [6] POPULAR *Počítačové vidění* [online]. [cit. 2017-02-10].
Dostupné z: <http://popular.fbmi.cvut.cz/optoel/Stranky/default.aspx>
- [7] Person Detection and Head Tracking to Detect Falls in Depth Maps. KEPSKI Michal, KWOLEK Bogdan. [online]. [cit. 2017-03-26]. Dostupné z:
http://galaxy.uci.agh.edu.pl/~bkw/research/pdf/2014/KepskiKwolek_ICCVG2014.pdf
- [8] Real-Time Human Pose Recognition in Parts from Single Depth Images . SHOTTON Jamie, FITZGIBBON Andrew, COOK Mat, SHARP Toby, FINOCCHIO Mark, MOORE Richard, KIPMAN Alex, BLAKE Andrew. *Microsoft Research Cambridge & Xbox Incubation* [online]. [cit. 2017-03-27]. Dostupné z: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf>
- [9] Alternativní rozhraní pro ovládání počítače. ČERNOBILA Ondřej. [online]. [cit. 2017-03-27]. Dostupné z: https://is.muni.cz/th/139763/fi_m/Diplomova_prace_Ondrej_Cernobila_FI_MUNI.pdf
- [10] Accurate 3D Pose Estimation From a Single Depth Image. YE Mao, WANG Xianwang, YANG Ruigang, REN Liu, POLLEFEYS Marc. [online]. [cit. 2017-03-27]. Dostupné z: <https://www.inf.ethz.ch/personal/pomarc/pubs/YeICCV11.pdf>
- [11] Tvorba 3D modelu řidiče na základě hloubkových map. FUCHSÍK Ondřej. [online]. [cit. 2017-03-27]. Dostupné z: <https://dspace.vsb.cz/handle/10084/115923>
- [12] 3D body reconstruction & recognition. *TraumaBot* [online]. [cit. 2017-02-10]. Dostupné z: <http://traumabot.blogspot.cz/2013/02/kinect-colour-ir-depth-image-reading.html>

- [13] Kinect v1 SDK C++. ZHANG, Edward. *Kinect Depth Data* [online]. [cit. 2017-02-10].
Dostupné z: <http://homes.cs.washington.edu/~edzhang/tutorials/kinect/kinect2.html>
- [14] Basic Thresholding Operations *OpenCV* [online]. [cit. 2017-02-12]. Dostupné z:
<http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>
- [15] Digitální zpracování a analýza obrazu. SOJKA, Eduard. [online]. [cit. 2017-02-16]. Dostupné
z: http://mrl.cs.vsb.cz/people/sojka/dzo/digitalni_zpracovani_obrazu.pdf
- [16] Background subtraction. *Wikipedia.org* [online]. poslední aktualizace 2017-02-08 [cit. 2017-
02-12]. Dostupné z: https://en.wikipedia.org/wiki/Background_subtraction
- [17] Background subtraction. *OpenCV* [online]. [cit. 2017-02-12].
Dostupné z: http://docs.opencv.org/3.1.0/db/d5c/tutorial_py_bg_subtraction.html
- [18] Operations on Arrays. *OpenCV* [online]. [cit. 2017-02-12].
Dostupné z: http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html
- [19] Smoothing images. *OpenCV* [online]. [cit. 2017-02-28].
Dostupné z: http://docs.opencv.org/3.2.0/d4/d13/tutorial_py_filtering.html
- [20] Computing moments and features. GAURA, Jan [online]. [cit. 2017-02-17].
Dostupné z: http://mrl.cs.vsb.cz/people/gaura/ano/features_02.pdf
- [21] Jak najít objekt, jehož šablonu mám k dispozici? *is.mendelu.cz* [online]. [cit. 2017-02-27].
Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=25216
- [22] Template Matching. *OpenCV* [online]. [cit. 2017-02-27]. Dostupné z:
[http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/
template_matching.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html)
- [23] Plane (geometry) *Wikipedia.org* [online]. poslední aktualizace 2017-01-26 [cit. 2017-03-09].
Dostupné z: [https://en.wikipedia.org/wiki/Plane_\(geometry\)](https://en.wikipedia.org/wiki/Plane_(geometry))
- [24] Histogram of oriented gradients *Wikipedia.org* [online]. poslední aktualizace 2017-03-03 [cit.
2017-03-19]. Dostupné z: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
- [25] Histogram of Oriented Gradients. *Learn OpenCV* [online]. [cit. 2017-03-19].
Dostupné z: <http://www.learnopencv.com/histogram-of-oriented-gradients/>
- [26] Support vector machines *Wikipedia.org* [online]. poslední aktualizace 2017-01-13 [cit. 2017-
03-19]. Dostupné z: https://cs.wikipedia.org/wiki/Support_vector_machines

A Obsah přiloženého CD

Struktura přiloženého CD:

- **/KOD** - v této složce se nachází projekt vytvořený pro detekci lidské kostry v hloubkové mapě.
- **/NAVOD** - zde se nachází soubor s postupem jak nainstalovat potřebné knihovny a následně spustit diplomovou práci.
- **/SPUSTITELNE** - spustitelná verze diplomové práce bez nutnosti instalace vývojového prostředí.
- **/TEXT** - elektronická podoba diplomové práce.
- **/VIDEOA** - natočená původní videa včetně detekce lidské kostry. Dále se zde nachází jednotlivé snímky z videa.